





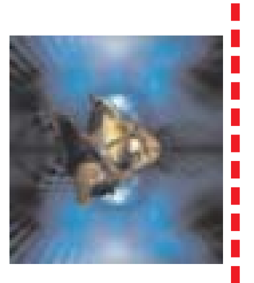
Kerberos

Este artículo pretende dar un entendimiento de Kerberos: para qué sirve, cómo funciona. La filosofía general detrás del funcionamiento de Kerberos.



Ovis Link

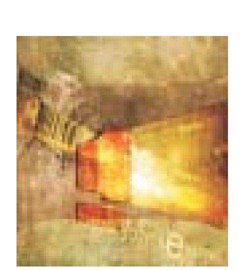
La empresa taiwanesa OvisLink nos ofrece un producto de excelentes prestaciones, un switch Gigabit Ethernet de 8 bocas de muy fácil implementación pero potentes características.



Debian

¿Por qué Debian? ¿Qué es Debian?

En este artículo contestamos estas preguntas y muchas otras más.



Hot Programming Languages

C/C++

C#.NET

Java

SQL

XML



editorial

Abril 2004.

Y seguimos creciendo.
Esta vez, en NEX7, ampliando las temáticas de NEX y 4 páginas más. En este "issue" inauguramos "Revisiones de Exámenes" analizando en detalle el examen LPI 101 de la certificación LPIC del Linux Professional Institute. Continuaremos en ediciones próximas con exámenes de otros vendors u organizaciones.
También volvimos a la carga con un tema ya desarrollado en NEX4: programación. El artículo "Lenguajes HOT de Programación" les permitirá tener una idea de a dónde apunta hoy el mundo de la programación. Esto es muy útil como conocimiento general pero también nos da el "trend" y nos orienta sobre en donde debemos poner nuestras energías. Muchas veces es importante conocer el "big picture" de las tendencias en un cierto área del conocimiento o del mercado para así tomar decisiones acertadas.
A continuación se detallan otros "highlights" de este "issue":
DFS (Distributed File System) nos muestra una nueva característica introducida por Microsoft en Windows 2000.
NEX7 presenta nuestro primer artículo sobre Debian. Es una introducción al porqué Debian y qué es Hurd. Es muy interesante conocer Debian ya que es la distribución que está adoptando el gobierno argentino como sistema operativo a integrar en las reparticiones públicas.
Nuestro "Suplemento de Seguridad" sólo tiene una temática desarrollada a fondo: Kerberos. Esto se complementa con los artículos en anteriores NEX: "Introducción a la Criptografía I y II". Ningún profesional en IT puede desconocer estas temáticas básicas.
Ya vamos por nuestro NEX Lab número 2 en donde pondremos a prueba: Switch GBit.
Otros pequeños artículos de interés complementan este número y siempre nos da una sonrisa el humor de Marcos Severi.

Nos vemos en NEX8.



Staff

Año 3 - Número 7 - Abril 2004

Director
Dr. Osvaldo Rodríguez

Propietarios
COR Technologies S.R.L.

Coordinador Editorial
Carlos Rodríguez Bontempi

Cordinación General
María Luján Zito

Responsable de Contenidos
Dr. Osvaldo Rodríguez

Editor en Jefe
Raúl Kuzner

Redactores
Martín Sturm, Javier Pierini, Raúl Kuzner, Osvaldo Rodríguez, María Luján Zito, Leonel F. Becchio, Rodrigo M. González, Hugo Cela, Guido Lorenzutti.

Humor
Marcos Severi

Distribución
Lorena De Lillo, Ximena Antona

Diseño Web Site
Emanuel A. Rincón

Diseño Gráfico
Víctor Pereyra
Carlos Rodríguez Bontempi

Publicidad
Ximena Antona
publicidad@nexweb.com.ar
+54 (11) 4312-7694

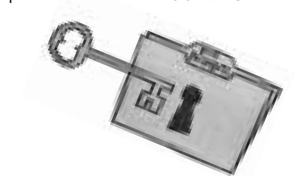
Preimpresión e Impresión
Talleres Gráficos S.A.
Buenos Aires Herald Ltd.Ind.Fin.
Azopardo 455
C1107ADE - Capital Federal

NEX - Periódico de Networking
Registro de la propiedad Intelectual en trámite leg3038

Dirección: Av. Córdoba 657, Piso 12
C1054AAF - Capital Federal
Tel: +54 (11) 4312-7694
http://www.nexweb.com.ar

El staff de NEX colabora ad-honorem, si desea escribir para nosotros, enviar un e-mail a:
articulos@nexweb.com.ar

Retire su ejemplar NEX en forma gratuita en Av.Córdoba 657, Piso 12 - Capital Federal
o solicítelo telefónicamente para su empresa al +54 (11) 4312-7694
http:// www.nexweb.com.ar





Lenguajes Hot de Programación

Las Software Development Conferences (SDC) buscan informar sobre las tendencias para el desarrollo de software. Hacia dónde apunta el mundo de los desarrolladores. La conferencia SDExpo que sucedió en Santa Clara (California, USA) hace unos pocos días les puede brindar un panorama muy completo sobre todo esto (ver www.sdexpo.com).

Clasificación del Software según su finalidad

Para poder distinguir y explicar los diferentes lenguajes que detallaremos en este artículo, haremos una división conceptual respecto del ambiente para el cual se desarrollan aplicaciones.

Cualquier clase de Software (Léase: Sistema Operativo, *Driver*/Controlador, Aplicación ó Servicio) a desarrollar, entra en una o varias de las siguientes clasificaciones:
- Software con acceso directo al Hardware y/o Dispositivos.
- Software a ejecutarse sobre una Plataforma de Software ya existente.

Esta última, destinada en su totalidad a Software Aplicativo, volveremos a dividirla conceptualmente haciendo las siguientes distinciones:
- Software para Productividad.
- Software para Acceso a Servicios.
- Software para Acceso a Datos.

Software con acceso directo al Hardware y/o Dispositivos

Esta clase de Software es la que esta destinada a monitorear y controlar el comportamiento de alguna clase de hardware y/o dispositivo específico. Ya sea que se trate de microcontroladores ó microprocesadores en sus diferentes clases, nombraremos a este software de "bajo nivel" o "de base".
El software de bajo nivel está presente en un sinnúmero de dispositivos en nuestra vida cotidiana: relojes, calculadoras, sistemas de alarma, seguridad y performance de automóviles (ABS, Inyección electrónica, etc.), teléfonos celulares, computadoras, impresoras, etc.

En IT (*Information Technology*-Tecnología de la Información), está clase de software también es conocido con varios nombres:
- *Driver*, cuando está destinado a controlar un periférico específico (Impresoras, Scanners, Placas de Video, Placas de Sonido, Placas de Red, etc.).
- Sistema Operativo, cuando está destinado a controlar una plataforma de hardware específico (Sparc, Mac, Intel y compatibles)

En estos momentos debe estar cruzando por su mente un pequeño dilema: "*Si un driver es específico para un Hardware. ¿Por qué hay versiones del mismo driver, para diferentes Sistemas Operativos?*" Sencillo, porque el *driver*, además de tener las instrucciones necesarias para dialogar con el hardware, tiene instrucciones específicas para los sistemas operativos de diferentes plataformas. De esta manera el software de base abstrae conceptualmente al hardware.

Software a ejecutarse sobre una Plataforma de Software ya existente

Esta clase de software es la que está destinada a depender exclusivamente de la existencia y ejecución previa de un software de base. Dentro de esta clase está todo el software que no fue hecho para un hardware específico.

Aquí, el software ya no depende ni tiene interacción alguna con el hardware, de hecho, todo proceso realizado por el software que tenga alguna clase de salida hacia o entrada desde un hardware particular no lo hará directamente, sino que lo hará a través del software de base.

También en esta clase está el software que permite la interaccción entre el software de base y el software específico para una tarea. ¿Muy confuso? Las siguientes clasificaciones aclararán este concepto.

Software para Productividad

El software de productividad tiene un alto grado de interacción con el usuario y esta diseñado para cumplir tareas específicas para los mismos. En esta clase podemos englobar a todos las aplicaciones que hacen alguna clase de tarea específica (a veces no tan específica) para que el usuario final pueda realizar algún proceso de productividad.

Podemos citar como ejemplo una aplicación de calculadora científica: se limita a recibir valores y operaciones de parte del usuario para luego devolver un resultado. Otro ejemplo son las aplicaciones que están contenidas en un "paquete de oficina", como Microsoft Office ú Open-Office.
Cada una de estas aplicaciones comparte características de funcionamiento que son comunes a todas: interactúan con el usuario, incrementan su nivel de productividad, y los datos que manipulan son almacenados en un formato particular que sólo ellas comprenden, así un archivo de una hoja de cálculo no puede ser "abierto" por un software de procesador de textos ni un software de presentación de diapositivas.

Software para Acceso a Servicios

En esta clase entran las aplicaciones desarrolladas específicamente para dar alguna teracción con el usuario. Por supuesto que cuando un software dialoga con otro software lo hace de manera específica, utilizando un protocolo estandarizado. Un ejemplo: las aplicaciones Cliente y Servidor de e-Mail. Usando este ejemplo particular vemos un software que está diseñado para interactuar con el usuario y además con otro software (que tampoco es software de base) y un software que está diseñado para interactuar solamente con otro software.

Pongamos el ejemplo en la práctica: usted redacta un e-mail a un amigo, usando un cliente de correo de su preferencia (Usuario <- a -> Software), cuando da la orden de enviar, el cliente de e-mail lo "envía" al servidor de e-mail de su Compañía ó Proveedor de Internet (Software <- a -> Software).

De este punto en adelante, hasta que el e-mail llegue a destino, el software pierde interacción con el usuario, solamente se limita a interactuar con otro software específico de traslado de e-mails (Software <- a -> Software).
Una vez en destino el e-mail solo podrá ser leído utilizando un cliente de e-mail (Software <- a -> Usuario).

La cantidad de implementaciones de software para acceso a servicios es muy amplia, citando ejemplos conocidos: Clientes de e-Mail (Outlook, Eudora, Evolution, etc.), Navegadores/*Browsers* de Internet (Internet Explorer, Netscape, Mozilla, Opera, etc.), Programas de Mensajería Instantánea (AOL Instant Messenger, MSN Messenger, Yahoo! Messenger, ICQ, etc.) Clientes de acceso a recursos ó información (FTP, NFS, SMB, DNS).

Software para Acceso a Datos

Aquí nos vemos en la obligación de hacer una distinción entre dos tipos de Software, el que va a solicitar acceder a los datos y el software que administrará, alojará y permitirá (o no) el acceso a esos datos.

El primero puede ser casi cualquier tipo de software desarrollado para tal fin (excepto el "de base").
Con el segundo nos referimos específicamente a los RDBMS (*Relational DataBase Management System*-Sistema de Administración de Base de Dato Relacional), cuyo objetivo primordial es almacenar de manera consistente, ordenada y uniforme los datos que serán accedidos por diferentes aplicaciones y/o servicios.

Una característica a resaltar de esta clase de software es que permite la simultaneidad de acceso, a los mismos datos, por varias aplicaciones y/o servicios.

Algunas Consideraciones

Si bien, con lo que expresamos hasta aquí, podemos considerar que esta clasificación es un poco caprichosa, aceptamos que deben existir varias maneras diferentes de clasificar el software, pero ésta es la que consideramos que permite una mejor clasificación de los lenguajes que citaremos más adelante.
También queremos hacer notar que mucho software presente en el mercado abarca más de una de las clasificaciones anteriores, e incluso algunos se comportan como excepciones. Un ejemplo: el RDBMS de Oracle permite a un administrador configurar el almacenamiento de bases de datos en un *Raw-Device* (literalmente: Dispositivo Crudo), accediendo directamente a un dispositivo físico sin intervención de ningún software de base (*Driver* ó Sistema Operativo), permitiendo de esta manera un incremento en la performance.

¿Qué lenguajes debemos/podemos usar?

Para cada uno de los lenguajes que citaremos a continuación, haremos una breve reseña histórica, mostraremos sus características principales y áreas comunes de utilización (según la división que realizamos anteriormente).

C++

Durante la década del 80, en los laboratorios *Bell*, *Bjarne Stroustrup* desarrolla C++ como una extensión al lenguaje C (originalmente desarrollado por *Dennis Ritchie* en los mismos laboratorios durante la década del 70). De todas las características que le fueron agregadas, la más notoria de todas, es la capacidad OOP (*Object Oriented Programming*-Programación Orientada a Objetos).
De esta manera, el código fuente es más reducido, más sencillo de entender, modificar y corregir. C++ puede ser considerado un lenguaje híbrido, debido a que permite programar estilo C, estilo OOP ó ambos.

El punto fuerte de C++ es la capacidad OOP

C++ provee una colección de clases predefinidas y la capacidad de generar nuevas clases por parte del programador. El conjunto de clases de C++, que son principalmente tipos de datos (*data types*), pueden ser instanciadas la cantidad de veces que sea necesario.

Las definiciones de clases, pueden contener tipos simples de datos (*data members*) y opcionalmente funciones (*member functions*). Las clases también pueden derivarse de una o más clases pre-existentes: ésta característica es llamada "herencia". Asimismo, los objetos

contenidos en una clase, “heredados” de una clase padre, pueden modificarse: ésta característica es llamada “polimorfismo”.

La suma de estas cualidades hace que C++ sea considerado un lenguaje clásico, vigente y buscado para el desarrollo de software de base, aunque no hay inconvenientes en que sea utilizado para desarrollar software Aplicativo.

Java

A principios de la década del 90, *Sun Microsystems* desarrolló el lenguaje *Oak*, destinado principalmente a pequeños dispositivos de entretenimiento y video a demanda. Debido a que ese mercado particular no evolucionó según lo esperado, *Sun Microsystems* decidió volcarse al emergente mercado de la WWW (*World Wide Web*). Desde ese momento en adelante el lenguaje es llamado *Java*.

Java posee características similares a las de C++: uno de los factores que desencadenaron la rápida adopción de Java como lenguaje de programación es su similitud de sintaxis con el lenguaje C++.
Es netamente Orientado a Objetos: **dentro de un programa Java todo es un objeto** y todo desciende por herencia de una clase “objeto raíz” (*root object class*).

El entorno Java incluye una gran cantidad de clases pre-existentes: de tipos de datos; de funciones utilitarias para manipulación de diferentes tipos de datos; de funciones de entrada-salida (*input/output*) para una gran variedad de orígenes de datos; de funciones de *networking* que permiten la comunicación entre aplicaciones sobre una plataforma de red y de funciones para desarrollar aplicaciones gráficas.

Una característica notoria es la de *Garbage Collection* (literalmente: Recolección de Residuos), que permite a los desarrolladores y programadores evitar los dolores de cabeza que produce la liberación explícita de memoria asignada dinámicamente.

Pero por sobre todas las cosas, la principal característica es su independencia del software de plataforma. Lo exitoso de esta funcionalidad es la separación conceptual de la ejecución de una aplicación: el código fuente Java no se compila para una plataforma específica de hardware, sino que se compila en un *byte-code* que será interpretado por una JVM (*Java Virtual Machine*-Máquina Virtual Java) o por un JIT (*Just In Time compiler*-Compilador en Tiempo de Ejecución).

De esta manera, el *byte-code* permanece universal para cualquier plataforma y la única porción de software específico para una plataforma son la JVM y el JIT.

Esta última característica sumada a las anteriores hace a Java ideal para desarrollar Aplicaciones *Cross-Platform* (multi-plataforma) y *Applets* (mini-aplicaciones) para Internet.

C#.NET

.NET es una iniciativa a gran escala que busca poner a Microsoft nuevamente como líder en innovaciones tecnológicas. Dejando de lado las consideraciones de marketing sobre la iniciativa “*Microsoft.NET*” para revolucionar la experiencia de los usuarios, es meritorio reconocer que Microsoft da un paso adelante en el concepto del análisis, codificación y utilización del software. Cuando creíamos que estaba todo dicho en el mundo del desarrollo de software, nos encontramos con un nuevo entorno de trabajo: el “*.NET Framework*”. Conceptualmente hablando, el *Framework* de Microsoft no es otra cosa que una implementación particular de la idea de software “independiente de la plataforma subyacente”.

Aquí se debe estar preguntando: “*¿Pero, acaso eso no es lo que hace la JVM de Sun?*”, la respuesta es: Sí. El *Framework* comparte casi todas sus características con la *JVM de Sun*, pero la característica que ideó Microsoft para comenzar a competir con *Sun* en el mercado de los desarrolladores de software *Cross-Platform*, es permitir que una gran cantidad de lenguajes de programación puedan ser compilados para poder ejecutarse sobre el *Framework*.

El proceso de compilación de estos lenguajes produce como resultado un CIL (*Common Intermediate Language*-Lenguaje Común Intermedio) que será interpretado, ó compilado en tiempo real, por el CLR (*Common Language Runtime*-Intérprete de Lenguaje Común). El CLR es uno de los componentes principales del *Framework*, el otro son las *Unified Class Libraries* (Librerías Unificadas de Clases).

La iniciativa “*Microsoft.NET*” tiene otro as en la manga: dentro de la suite de desarrollo “Visual Studio .NET” aparece el nuevo Lenguaje C# (pronunciado *see-sharp*)

Microsoft espera tener éxito en la adopción de este lenguaje por parte de programadores de C++ que todavía no adoptaron *Java*, ya que C# posee características similares a C++ y muchas características similares a *Java*.

Debido a la existencia del *Framework* y sus características, el lenguaje **C# es ideal para desarrollar Aplicaciones *Cross-Platform*** (multi-plataforma) y *WebServices* (servicios de acceso web) para Internet. .NET esta a la vuelta de la esquina para todos los desarrolladores que ya tienen experiencia previa en los lenguajes visuales de Microsoft y tiene un futuro muy prometedor.

XML

HTML (*HyperText Mark-up Language*-Lenguaje de Marcadores de Hipertexto) y XML (*eXtensible Mark-up Language*-Lenguaje de Marcadores Extensibles) comparten un padre en común, el SGML (*Standard Generalized Mark-up Language*-Lenguaje Estandarizado de Marcadores Generalizados). SGML es principalmente usado para el procesamiento de textos, HTML es una versión simplificada del SGML y XML es una versión filtrada del SGML.

Tanto HTML como XML adoptan la sintaxis de marcadores del SGML. Cada marcador, de inicio y fin, tiene como característica notoria estar “encerrado” entre signos “<” y “>”. Para distinguir el marcador de fin, éste tiene una “/” antes del nombre del marcador, por ejemplo “<B>” y “</B>”. Cada par de marcadores de inicio y fin, establece una característica particular al texto que está contenido entre ellos.

XML permite definir una estandarización propia para el intercambio de información

HTML tiene un juego de marcadores predefinidos que están dedicados exclusivamente a la interpretación estética de los contenidos de un documento, de esta manera los marcadores establecen el tamaño y distribución del texto, si es título, párrafo o lista, si debe ir en **negrita**, *itálica* o subrayado, indicadores de origen de imagen e indicadores de salto a otro texto (comúnmente llamados *links*). En la actualidad, el único uso que se le da al HTML es la difusión de documentos e información en la WWW.
XML no tiene ningún juego de marcadores predefinidos, permitiendo a las aplicaciones definir la semántica y estructura jerárquica de los marcadores.

Aunque esta característica aparentemente lo haga un lenguaje “débil”, justamente sucede todo lo contrario: las empresas son las que definen su propia estandarización para el intercambio de información y adaptan al XML para su propio beneficio.

SQL

Durante la década del 70, en los laboratorios de *IBM*, se comienza a desarrollar la teoría de bases de datos relacionales y durante el desarrollo de los RDBMS, crearon un cuasi-lenguaje para operarlos: el SEQUEL (*Structured English QUEry Language*-Lenguaje de Consultas Estructuradas en Inglés). Más adelante pasó a llamarse SQL (*Structured Query Language*- Lenguaje de Consultas Estructuradas) pero se lo sigue pronunciando “sequel”.
Ya en la década del 80, la empresa “*Relational Software Inc*” (hoy en día *Oracle Corporation*) popularizó la primera implementación comercial de SQL. Aunque cada desarrollador de un RDBMS tenga su propia implementación, SQL es aceptado como el lenguaje estándar de todos los RDBMS.

Los objetivos de SQL son: la creación y manipulación de tablas; inserción, actualización y borrado de datos; encontrar, filtrar y agrupar y totalizar datos; combinar consultas y demás funciones de manipulación de datos.

Con la ayuda de ODBC (*Open DataBase Connectivity*-Conectividad Abierta de Bases de Datos) un programador puede desarrollar una aplicación (*Front-End*) capaz de acceder a grandes volúmenes de datos contenidos en cualquier clase de almacenamiento RDBMS (siempre y cuando éste sea también compatible con ODBC) y de esta manera permitirle a un usuario final manipular los datos contenidos en las tablas del RDBMS.

Conclusión

Si bien mencionamos dos lenguajes que tienen una larga historia en el mercado (C++ y SQL), es notorio que los lenguajes que hoy en día están “Hot” son los que están orientados a desarrollar aplicaciones *Cross-Platform* (Java y C#). También se nota una marcada tendencia hacia el desarrollo de aplicaciones que interactúan entre sí, promoviendo cada vez más el uso de XML.

Raúl Kuzner
Analista en Sistemas



Direccionario

C/C++
http://cplus.about.com
Java
http://java.sun.com
C#.Net
http://msdn.microsoft.com/netframework/
http://msdn.microsoft.com/vcsharp/
XML
http://www.w3.org
http://www.xml.com
SQL
http://www.sqlcourse.com
http://www.oracle.com

¿Qué factores hacen “hot” a un Lenguaje?

La respuesta se reduce a: “Un lenguaje es *hot* debido a que hay muchos programadores que tienen interés en programar en él”. Siguiendo esta premisa, debemos considerar que los factores que “*hacen hot*” a un lenguaje no necesariamente son los mismos que “*mantienen hot*” a un lenguaje. Veamos algunos factores:

Simplicidad de uso: ésto parece ser muy importante en la elección de un nuevo lenguaje. Siempre hay que cuestionarse si un lenguaje será más rápido y simple de usar que otros lenguajes. Los paradigmas de programación y los nuevos lenguajes cambian lo que consideramos “simplicidad de uso”.

Características: un lenguaje puede ser elegido debido a que tiene alguna característica particular. Por ejemplo, si su objetivo es programar una aplicación que se ejecute desde Internet, su opción deberá ser Java o C# y no COBOL.

Performance: dependiendo del tipo de aplicación a desarrollar, la performance del producto final tiene mucho peso. Aquí vemos el porque C y C++ se mantendrán “hot” por mucho tiempo. También podemos apreciar otro punto de vista: ¿Por qué es necesario desarrollar aplicaciones interactivas en un lenguaje de alta performance?

Soporte Empresarial: aquí debemos tener en cuenta si existe alguna empresa ú organización que está “empujando” un software a que sea *hot*. Por ejemplo: “Si *Unix* no hubiese sido programado en C. ¿Sería C tan *hot*?” ó “¿La permanencia de *Basic* en el mercado puede deberse exclusivamente al soporte que *Microsoft* le da?”

Programadores experimentados: ésto es especialmente en proyectos a largo plazo. ¿Serán los futuros programadores capaces de entender el código existente y continuar con su desarrollo?

Código ya existente: la cantidad de código ya existente dicta la necesidad de programadores que entiendan dicho lenguaje. Este factor por sí solo puede “*mantener hot*” un lenguaje. El mejor ejemplo es COBOL: la única razón por la que es necesario que existan programadores es la gran cantidad de código y aplicaciones que necesitan mantenimiento.



El Nuevo Office System 2003

Acostumbrados como estamos a entender que una suite o conjunto de herramientas de trabajo de oficina consta básicamente de un procesador de texto, una hoja de cálculo y un sistema de presentaciones, sumados a un programa de correo electrónico y, en algunos casos, a un administrador de información personal, ahora debemos suponer que esto es solo lo más elemental con las que una de estas suites debe contar.

De hecho, bajo la nueva estrategia de Microsoft con la presentación de Microsoft Office System 2003, ésta sería la punta del iceberg de toda la serie de programas, aplicaciones para servidores y servicios adicionales que ahora contiene su suite de oficina, los cuales vienen empacados en diferentes configuraciones para cumplir con las necesidades de diferentes escenarios de trabajo.

Estas nuevas versiones de la tradicional suite se denominan, en general, Microsoft Office 2003 Editions, dentro de las que Microsoft ha incluido sus productos para que se acomoden a las necesidades ya no de los usuarios individuales, sino de los diferentes entornos corporativos que puedan aprovechar al máximo las novedades contenidas en este conjunto. Microsoft Office 2003 Editions cubre desde una edición básica hasta una profesional de tipo empresarial, pasando por opciones para el sector educativo y también para medianos y pequeños negocios, pero éstas las veremos más adelante.

En la nueva suite se pueden hallar hasta 22 productos diferentes, entre programas, aplicaciones para servidores y servicios adicionales, los cuales forman el nuevo concepto denominado Microsoft Office System 2003. Estas soluciones ya no se centran simplemente en facilitar los procesos individuales, como la realización de cartas y texto, o la producción y el análisis de cálculos; ahora Microsoft pretende dirigirse hacia los procesos que se realizan en la oficina, aprovechando así al máximo la información existente, no sólo en los documentos físicos o digitales, sino también en las iniciativas, ideas y conocimiento de los empleados para aumentar la capacidad productiva de la empresa. Esta iniciativa, que podría verse como algo idealista, se basa en el fortalecimiento de cuatro puntos esenciales dentro de las empresas, como son la comunicación, la colaboración, el manejo de la información y, por último, su transformación en resultados.

Esta iniciativa se basa en cuatro puntos esenciales: comunicación, colaboración, manejo de información y resultados.

Los principales cambios que sus usuarios podrán encontrar en estas nuevas versiones de sus tradicionales programas de escritorio serán su interfaz más moderna y con mayores posibilidades de personalización. Excel y Word no sufren muchos cambios en realidad, mientras que Power Point se hace más interactivo y fácil de manipular para realizar presentaciones de multimedios que incluyan audio y vídeo.

La más elemental de las versiones de esta suite de herramientas es la denominada Microsoft Office Basic Edition 2003, que se podrá encontrar en PCs nuevas exclusivamente y constará de las nuevas versiones de Word 2003, Excel 2003, Outlook 2003.
Junto con esta versión, aparece Microsoft Office Student and Teacher Edition 2003, dirigida hacia el segmento educativo y el hogar principalmente, pues como requisito para su adquisición se debe ser estudiante o maestro. En esta versión sólo se podrán encontrar las nuevas versiones de Word 2003, Excel 2003, Outlook 2003 y Power Point 2003.

Los usuarios de Microsoft Office Student and Teacher Edition 2003 realmente no notarán mayores cambios en sus aplicaciones e incluso los actuales usuarios de Office XP podrán vivir sin problemas en sus hogares sin necesidad de actualizarse a 2003.
Tal vez, la más interesante oferta de Microsoft frente a esta versión básica es la posibilidad de ser instalada hasta en tres computadores. El precio de esta versión también resulta el más bajo con relación a las demás, pues en Estados Unidos se ofrece a U$S 149 y el usuario no puede hacer actualizaciones de productos anteriores como Office 97 o XP.

La versión Microsoft Office Standard 2003 consta de los mismos programas que la anterior, pero dirigida hacia el mercado del profesional independiente, el hogar y las pequeñas oficinas. Incluye los mismos programas que la edición para el sector educativo, pero puede adquirirse la actualización para versiones anteriores.
Sobre esta versión, encontramos la edición para pequeñas y medianas empresas. En esta edición de Office System 2003, comenzamos a ver los primeros grandes cambios en cuanto a herramientas de productividad, pues aparte de los productos tradicionales, aparece Microsoft Publisher 2003, una versión más completa que sus predecesores y con características muy atractivas para realizar volantes, cuadernillos y hojas para impresión de material con características profesionales. Publisher 2003 también incluye patrones preconfigurados para facilitar el trabajo de los usuarios novatos y su interfaz ha cambiado para hacerse más sencilla y fácil de manejar. Por otro lado, esta nueva versión incluye nuevas herramientas como el Business Contact Manager, que se integra a Outlook 2003, con lo cual se convierte en un sencillo pero eficiente utensilio tipo CRM para hacer seguimiento a clientes actuales y potenciales.

Con Office Edición Professional 2003, Microsoft complementa la oferta de entrada. Como la edición para pequeños negocios, cuenta con Word 2003, Excel 2003, Power Point 2003, Publisher 2003 y Outlook con la herramienta Business Contact Manager. La adición de Access 2003 a este conjunto de herramientas le confiere al usuario la opción de manejar una base de datos y hacer más confiable la administración de información de proveedores, clientes y contactos.
Pero es con esta edición con la que comienzan a notarse los grandes cambios en cuanto a colaboración e interactividad que Microsoft desea promover dentro de los usuarios empresariales, gracias a la que la suite Office Edición Professional 2003 cuenta con la posibilidad de aceptar esquemas y formas en lenguaje XML, así como de crear y manejar contenido en formato IRM (siglas de Information Rights Management, una nueva serie de herramientas de seguridad para proteger documentos e información valiosa al interior de la empresa, compatible exclusivamente con los nuevos productos de Microsoft Office System 2003).
Con estas dos opciones, esta suite se convierte en el primer paso para la utilización de servicios adicionales no sólo de Microsoft sino de otras empresas que los ofrezcan a través de Internet, lo que genera mayores posibilidades de comunicación y de realización de procesos dentro de la empresa, así como con clientes y proveedores.

Desarrollo corporativo

Finalmente, sobre esta opción aparece la más importante transformación que Microsoft pueda ofrecer en la actualidad. Se trata de Microsoft Office Professional Enterprise Edition 2003, la cual en realidad es la respuesta a muchas tendencias tecnológicas actuales y que pretende convertirse no sólo en el conjunto de herramientas más utilizado en el mundo porque en realidad, ya lo es sino desarrollar una nueva manera de manejar la información.
Para comenzar, esta edición no se ofrecerá “en cajas”, sino a través de licencias por volumen a entes empresariales de gran tamaño. De esta forma, Microsoft no sólo planea actualizar sus usuarios de gran escala, sino que también pretende aumentar el volumen de su participación en el mercado de la colaboración y la interactividad.

Aparte de las aplicaciones ya mencionadas como World 2003, Excel 2003, Power Point 2003, Outlook 2003, Publisher 2003 y Access 2003, que pueden utilizarse de manera individual en cada escritorio y que en realidad sus cambios en cuanto a funcionamiento no son tan perceptibles a primera vista por el usuario, Microsoft Office Edición Professional Enterprise 2003 cuenta con una nueva herramienta denominada InfoPath 2003, cuyo nombre de código fue XDocs, basada en XML y que permite la creación de formularios para compilar y compartir datos, lo que facilita la interacción de los usuarios entre sí y con la información existente en los servidores de la empresa, sin importar si ésta procede de documentos Word o Excel o si está basada en Internet o en correo electrónico.
Otra novedad es SharePoint 2003, la herramienta de Microsoft para desarrollar portales corporativos, que ha mejorado sus



capacidades para administrar, configurar y desplegar información de Interés tanto de uso personal como empresarial. SharePoint 2003 también se hace más seguro con la aplicación de nuevas normas de acceso de usuarios, directorios y contenidos.

Mejoras y Novedades

Otros productos se pueden adquirir alrededor de Office System 2003 con el fin de aprovechar al máximo el avance tecnológico.
La tecnología denominada Grid Computing o computación según demanda, la movilidad derivada del incremento de las redes inalámbricas, y estas dos opciones unidas a una capacidad cada vez mayor en los procesadores que hacen más potentes, económicos y eficientes a los computadores (en especial a los equipos portátiles y las TabletPCs), permite colaboración e interactividad bajo diferentes mecanismos que van desde el simple intercambio de información a través de correo electrónico, hasta la búsqueda y el seguimiento de material de trabajo en bases de datos, aparte de permitir comunicación con diferentes medios, como mensajería instantánea y video conferencia.
Además, permite el uso de voz, vídeo y hasta mano alzada para procesar, almacenar y, por supuesto, intercambiar información.
Para esto, Microsoft ha presentado nuevas versiones de algunos de sus productos y ha presentado otros completamente nuevos, así como nuevos servicios y soluciones que complementan la oferta de Microsoft Office System 2003. Entre ellos podemos contar los siguientes:

- ***Microsoft Office OneNote 2003***. Probado inicialmente en las actuales TabletPCs, la nueva versión de OneNote pretende enriquecer la manera como los usuarios acceden a sus datos y a su información. En realidad, este producto es la manera como Microsoft supone que se deben tomar notas, recopilar datos y procesar información en el futuro más cercano. Con un fondo parecido al de un bloc de notas tradicional, que sirve sólo para evocar su función, OneNote 2003, puede recopilar gráficas, textos y gráficos realizados tanto con teclado y ratón como en mano alzada (aprovechando las características que aprendió en las TabletPCs) y transformarlos en texto digital, mientras almacena voz, sonido y vídeo para convertirlos posteriormente en información utilizable para realizar informes, textos o para compartirla por correo electrónico o publicarla en un sitio Web.

- ***FrontPage 2003***. Ahora este producto se aleja de su imagen compleja y torpe para ofrecer los resultados que el usuario desea al publicar un sitio Web. Incluye nuevos patrones de publicación y funciones enriquecidas, manejo de capas, diseño en JavaScript sin necesidad de programar y creación de gráficas y animaciones utilizando diferentes aplicaciones, así como la comprobación del contenido en diferentes navegadores para permitirle al diseñador mayor confiabilidad en los resultados deseados.

- ***Microsoft Exchange Server 2003***. Para el intercambio y la actualización de información es la versión más evolucionada de este programa que puede ser adquirido en dos versiones, una estándar y otra más robusta para ambientes corporativos.
Cuando este producto se integra con la suite de oficina a través de Outlook, Microsoft no incorpora su producto Business Contact Manager, pues este cumple con sus funciones, mientras que aumenta el caché de usuario para ofrecer mejores resultados en la actualización y desarrollo de actividades compartidas.

- ***Microsoft Office Project 2003***. Y su respaldo en servidor, para el manejo de actividades y proyectos corporativos compartidos con diferentes áreas de la empresa, empleados y aliados con el fin de hacerle seguimiento a diferentes actividades y procesos.
De paso, esta aplicación puede servir para medir la eficiencia de equipos de trabajo, personas, proyectos y procesos, facilitando así la administración y el diagnóstico del comportamiento de la empresa.

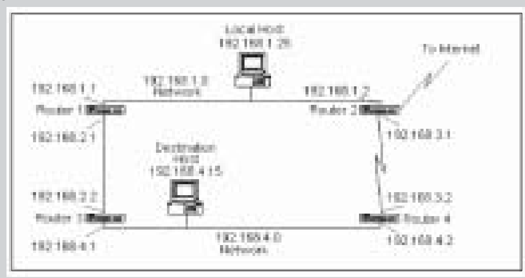
Por Marcelo Hernáez
Instructor Microsoft Office (MCT)



Pregunta Microsoft

Examen 070-293 - Planning and Maintaining a Microsoft Windows Server 2003 Network Infrastructure

Usted usa una computadora con Windows XP en una red de Windows 2003. La red está representada en el siguiente esquema:



La dirección IP de su computadora es 192.168.1.25. Porque el Router 2 es su conexión a Internet, su puerta de enlace predeterminada (default gateway) es 192.168.1.2.
Router 2 se conecta con Router 4 usando una conexión dial-up; por lo tanto, quiere asegurarse que su computadora siempre routee información a través de Router 1 en vez de Router 2 cuando mande datos a la red 192.168.4.0, al menos que el Router 1 esté inhabilitado. Usted quiere mantener esta configuración de rutas independientemente de cuántas veces reinicie su computadora.

¿Cuál de los siguientes comandos debería tipear en el command prompt para agregar las apropiadas rutas al host de destino?

```
A.     route add 192.168.4.0 mask 255.255.255.0 192.168.1.1 metric 1
route add 192.168.4.0 mask 255.255.255.0 192.168.1.2 metric 2
B.     route f add 192.168.4.0 mask 255.255.255.0 192.168.1.1 metric 1
route f add 192.168.4.0 mask 255.255.255.0 192.168.1.2 metric 2
C.     route f add 192.168.1.1 mask 255.255.255.0 192.168.4.0 metric 1
route f add 192.168.1.2 mask 255.255.255.0 192.168.4.0 metric 2
D.     route p add 192.168.1.1 mask 255.255.255.0 192.168.4.0 metric 1
route p add 192.168.1.2 mask 255.255.255.0 192.168.4.0 metric 2
E.     route p add 192.168.4.0 mask 255.255.255.0 192.168.1.1 metric 1
route p add 192.168.4.0 mask 255.255.255.0 192.168.1.2 metric 2
```

Respuesta Correcta: **E**

Curso 2285: **Installing, Configuring, and Administering Microsoft Windows XP Professional**

Gratuito para Empresas*

COR Technologies SRL ofrecerá durante los meses de Abril y Mayo el primer curso de la carrera MCSA y MCSE de forma gratuita a todos aquellos interesados.

Course 2285: Installing, Configuring, and Administering Microsoft Windows XP Professional (16 hs) (Primer Curso de la Carrera MCSA y MCSE Windows 2003).

Promoción válida para Grandes Empresas u Organizaciones (con Facturación Anual mayor a un Millón de Pesos) y cualquier Organismo Público Municipal, Provincial, o Nacional.

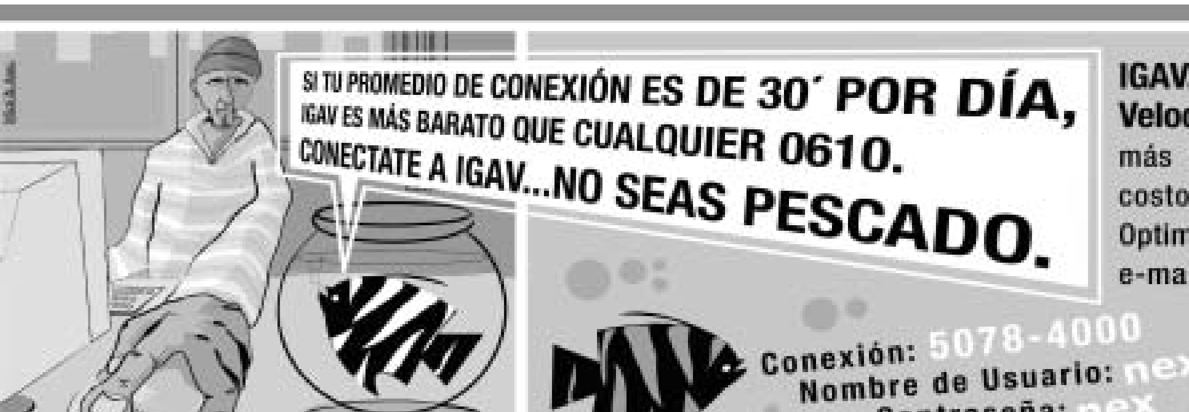
Para más información enviar email a microsoft@cortech.com.ar

Especificar en el mismo:

- Razón social de la Empresa
- CUIT
- Dirección Postal
- Nombre de Responsable de Contacto
- Telefono de Responsable
- Email de Responsable
- Cantidad de Personas que realizarían la Capacitación.
- Horarios y fechas disponibles para realizarla.

Luego se comunicará un asesor de Ventas con Ustedes para coordinar el grupo de participantes a los horarios disponibles.

*Las empresas que se postulen para recibir este beneficio deberán ser previamente calificadas y seleccionadas por COR Technologies SRL para participar en la promoción. Los participantes de la misma deberán abonar solamente el Material Microsoft Oficial (MOC). No hay máximos de alumnos que puede enviar cada Empresa. Promoción sujeta a disponibilidad de vacantes y hasta agotar Stock.





THE DISTRIBUTED FILE SYSTEM (DFS)

¿De qué se trata DFS? ¿Cuáles son sus beneficios y principales características? En los siguientes párrafos podrán responderse todas estas cuestiones de suma importancia para el trabajo diario.

Para empezar....

Con DFS, se pueden crear en una sola estructura jerárquica, todos los recursos compartidos de una red de trabajo.
Bajo una raíz principal, se pueden crear vínculos que apunten a todos los recursos compartidos a través de numerosos servidores, realizando un vínculo por cada recurso compartido. Usando DFS para compartir recursos, los usuarios sólo necesitan acordarse un solo lugar dónde acceder a esos recursos, sin importar dónde se encuentren alojados.
DFS presenta mayor utilidad en grandes empresas y probablemente no valga el esfuerzo en redes de pequeñas empresas.
Luego de la configuración de la consola de DFS, se tendrá una sola raíz DFS con todos los recursos compartidos de la organización listados en ella. Todos esos recursos, no importa dónde se encuentren físicamente son mostrados bajo la raíz principal. Los vínculos a los recursos son vistos por los usuarios como sub-directorios de esa raíz principal. Los usuarios no deben ver, ni preocuparse, ni siquiera saber qué servidor es apuntado por cada uno de esos sub-directorios. Tampoco necesitarán saber cuántos servidores están involucrados.
Sin embargo hay que dejar en claro que DFS no es una clase de servidor de archivos. DFS no crea los recursos compartidos si no que éstos deben ser creados primero en los servidores correspondientes y luego utilizar DFS para imponerles un cierto orden.
Con respecto a este tema también es importante destacar que no hace falta que esos recursos compartidos estén en Windows NT, 2000 o Server 2003, tranquilamente DFS puede usarse con Unix o Novell Netware.
Por otro lado al combinar DFS con las bondades de Active Directory, la raíz DFS puede ser hecha tolerable a fallos. Esto es posible gracias a que esa raíz puede ser guardada en Active Directory, con lo cual será mantenida a través de todos los controladores de dominio. Entonces, si uno de los servidores conteniendo recursos compartidos falla, los usuarios son redireccionados a otro servidor que contenga información de esos recursos sin que ellos noten ningún cambio.La pregunta entonces es, ¿se puede de alguna manera proteger esos recursos compartidos y su información con alguna clase de tolerancia a fallos? Sí, con vínculos con tolerancia a fallos. Si se tienen 2 recursos idénticos para compartir, entonces se pueden configurar esos recursos con DFS en conjunción con FRS (File Replication System) para replicar la información existente en esos recursos asegurando su consistencia.Usando FRS, DFS puede mantener todas las copias de los destinos replicados en sincronía entre todos.

Terminología DFS

Un montón de nuevos conceptos y términos han aparecido. Empecemos por el de raíz. Esta se traduce directamente en el recurso que será visible a la red de trabajo. Se pueden tener muchas raíces en un site, y con Windows Server 2003 un servidor puede mantener más de una raíz, lo que antes era una limitación en Windows 2000.
Bajo una raíz se pueden agregar vínculos de DFS. El vínculo es otro recurso compartido en algún lugar de la red de trabajo que es ubicado debajo de la raíz.
Los vínculos dentro de la arquitectura jerárquica de DFS son como hipervínculos en una página Web que automáticamente redireccionan al usuario a una nueva locación.
Un target o réplica puede referirse tanto a una raíz como a un vínculo.
Si se tienen 2 recursos compartidos idénticos en la red de trabajo, usualmente en servidores separados, se pueden agrupar juntos en el mismo vínculo como targets de DFS. Se pueden replicar también raíces enteras como un miembro de réplicas de la raíz.

Stand-Alone versus Fault tolerant

Antes de comenzar, es necesario decidir qué clase de DFS se necesita o se quiere realizar. La gran diferencia va a estar en la raíz del DFS.
En un DFS basado en Active Directory, o DFS con tolerancia a fallos, la raíz por sí misma puede tener réplicas. De esta forma esa raíz puede ser distribuida por todos los servidores de Active Directory y así lograr la tolerancia a fallos.
Además cuando DFS está integrado con AD las réplicas de los vínculos pueden ser configuradas para usar replicación automática.
Con esta característica, FRS (File Replication Service) se encarga de la sincronización de los contenidos de las carpetas replicadas para asegurar que todas las réplicas contienen la misma información.
Por último y no menos importante, con DFS basado en AD no solo los usuarios no necesitan saber en qué servidor se encuentra determinado recurso compartido sino que ni siquiera saben en qué servidor se encuentra el DFS activo.
Un DFS basado en dominio automáticamente publica su topología en el AD, lo cual significa que su estructura jerárquica es publicada en AD para que todos los controladores de dominio sepan dónde reside DFS.
Pero ojo, esto no significa que cada controlador de dominio sea una réplica de la raíz.
Por todas estas razones es que cuando se tiene un dominio basado en Active Directory se debe elegir DFS con tolerancia a fallos.
¿Pero qué pasa en el caso de no contar con Active Directory?
Un DFS stand-alone es un paso sólido hacia el mundo de Windows 2000 o Windows 2003 Server, sin requerir una estructura de Active Directory.
Con un DFS stand-alone, no se tiene la linda tolerancia a fallos por sí misma de la raíz, o la replicación automática, o la publicación de DFS en AD. Pero aún se pueden obtener todos los otros beneficios que otorga DFS.

Creando una raíz DFS

Como vimos anteriormente las dos opciones para el tipo de raíz de DFS son, raíz de dominio o raíz stand-alone. Una raíz de dominio va a publicarse por si misma en el Active Directory, mientras que la raíz stand-alone no lo hace.
Esta diferencia fundamental es el factor decisivo para saber cuánta funcionalidad se va a recibir. Hay que tener en cuenta que una raíz DFS de dominio debe ser alojada en un controlador de dominio, esto significa que existe un AD con el cual trabajar.
Una de las ventajas principales es que las raíces de dominio pueden tener réplicas, lo cual permite que cualquier controlador de dominio pueda hostear la raíz, aumentando de esta forma la tolerancia a fallos. El próximo paso es elegir un dominio el cual alojará al DFS.
El propósito detrás de la selección de un dominio para la raíz es publicar el DFS en AD.
Esto nos lleva a otra ventaja de usar raíces con tolerancia de fallos, que permitirá que los usuarios nunca tengan que preocuparse acerca de qué servidor aloja a cada recurso.
Luego de seleccionar el dominio, debe elegirse el servidor que va a hostear la raíz, y además se debe proveer de un nombre de raíz y una descripción para la misma.
El nombre para cada raíz debe ser único en el dominio.
A continuación se debe definir el recurso compartido para esa raíz. Este recurso puede estar ya compartido o se puede compartir en este momento.

Agregando vínculos a la raíz

Una vez que se tiene una raíz, es necesario agregar los vínculos de DFS de esa raíz. Debe ponerse un nombre al vínculo para usarse como referencia dentro de la estructura DFS, y luego el nombre del recurso compartido existente al cual se está vinculando.
Una característica interesante que puede configurarse es la cantidad de tiempo que la referencia al recurso permanece en memoria caché, la cual nos indica cuánto espera el sistema para actualizar la información del recurso compartido.
En caso de que un vínculo se encuentre offline el cliente va a ser informado en ese tiempo y va a ser redirigido a un recurso apropiado, de acuerdo a la existencia o no de réplicas.
Hay que pensar bien el valor de esta característica, dado que cuanto más corto es el intervalo habrá mayor tráfico en la red, y cuanto más largo, mayor el tiempo para realizar la actualización.

Configurando las réplicas

Aquí primero debe elegirse el vínculo que quiere replicarse, luego escribir el camino para el próximo recurso que va a estar disponible como parte de este vínculo, y chequear dónde se va a agregar el target de la replicación.
El FRS en Windows 2003 Server trabaja junto con DFS para sincronizar los archivos en los recursos incluidos en el vínculo. Algunas mejoras de Server 2003 en lo que respecta a FRS incluyen la compresión del tráfico de replicación y la habilidad de diferenciar entre el tráfico de replicación necesario e innecesario.
Uno de los recursos compartidos que se ha configurado en la replicación debe ser iniciado como maestro para que comience el proceso de replicación. Esta asignación como maestro es solamente temporal a los fines de comenzar con la replicación.
La replicación para DFS ha mejorado con la tecnología .NET, y ahora se puede elegir entre varias topologías disponibles o incluso crear una. De esta forma se finaliza el proceso de replicación haciendo disponibles varios recursos similares para los usuarios de DFS.
Además se ha configurado la replicación entre los recursos, de manera que las nuevas aplicaciones solamente deban ser agregadas en uno de los recursos, replicándose automáticamente en los otros.
Por otro lado se puede reconfigurar la topología y la programación de la replicación en cualquier momento. Incluso se puede decidir qué archivos o subcarpetas en el recurso compartido son replicados.
A partir de aquí cuando los usuarios quieran conectarse con un recurso, serán redirigidos a cualquiera de las réplicas de ese recurso.
En caso de querer remover una réplica puede hacerse libremente sin alterar el normal funcionamiento de DFS.

Entendiendo la replicación

En un DFS stand-alone la replicación es manual y una de las réplicas del vínculo es el maestro, lo cual significa que los cambios en el maestro se propagan a todas las otras réplicas.
Si el recurso físico que quiere mantenerse sincronizado reside en un volúmen NTFS en Windows 2000 Server, la replicación es automática y se usa replicación multi-maestro, con lo cual se pueden modificar archivos en cualquiera de las réplicas del vínculo, y los cambios son automáticamente copiados.
Para poder administrar las réplicas primero debe verse la información de la replicación en la consola de manejo de DFS.

Administrando DFS

Luego de haber configurado DFS hay una serie de pasos que realizar para poder administrar las raíces, los vínculos y los clientes conectados a ellos.
Cuando se tienen varios recursos replicados se pueden encontrar ocasiones en las cuales colocar alguno de ellos offline, por ejemplo cuando se debe realizar el mantenimiento de un servidor. En estos casos sólo es necesario poner offline el recurso perteneciente a ese servidor sin que los usuarios pierdan sus conexiones, seleccionando Enable/Disable a la referencia. Por otro lado, periódicamente, debe verificarse el estado de cada uno de los vínculos dentro de la topología DFS, solamente haciendo un click en Check Status en la barra de herramientas de la consola. Un nodo con fallas indica que hay un problema al acceder a la carpeta compartida a la cual el nodo hace referencia. En caso de ser necesario puede removerse el nodo fallido sin que los usuarios se vean perjudicados, ya que serán redirigidos a otro recurso de la réplica. Además dentro de la consola de administración de DFS un usuario se puede conectar con cualquier raíz DFS dentro de la red de trabajo, con sólo hacer click en Show Root. A partir de ahí se puede navegar por la red buscando raíces o tipear la ubicación de las mismas. Para remover una raíz DFS de la consola de administración solamente es necesario seleccionar Delete Root con el botón derecho del mouse.
Antes de configurar la raíz, establecer vínculos y reorganizar la forma en que los usuarios acceden a los recursos de la red debemos recordar que DFS no se trata de jugar con nuevas características de Windows, sino de hacer la vida más fácil.

... Por último

Debemos saber que DFS es una buena forma de consolidar los recursos de una empresa. Se pueden tomar todos los recursos compartidos a través de la red de trabajo y ponerlos bajo un solo recurso lógico. Lo importante del tema es que actualmente configurar DFS no tiene absolutamente ningún impacto en la configuración de la red de trabajo.Se puede preparar toda la configuración de DFS sin que nadie siquiera lo note, pero una vez terminado el trabajo de configuración la parte difícil comienza, cambiar los mapeos actuales de los usuarios, uno por recurso de cada servidor, por un solo mapeo para todos los recursos. La buena noticia con DFS es que uno se puede olvidar de lidiar con los mapeos de las unidades. Además ahora cuando se desee reemplazar un servidor por otro nuevo, se puede configurar el nuevo como un vínculo offline hasta asegurarse que cuente con toda la información necesaria actualizada y a partir de ese momento poner online el nuevo servidor y dar de baja al otro, sin siquiera hacer un backup del mismo.

Rodrigo González

NEX - LAB - NEX - LAB
LABORATORIO
Swicht OvisLink Live-GSH8T

En la actualidad, las LAN están cada vez más congestionadas, producto de una población de usuarios en constante crecimiento y con aplicaciones que requieren mayor un tráfico de información. Esto desencadena la necesidad de un mayor ancho de banda o bien un mejor aprovechamiento del existente. El switch juega un papel importante para subsanar inconvenientes inherentes a este tema.

Ante todo debemos definir qué es un switch, cómo trabaja y para qué necesitamos tener uno en nuestra red. Si nos remitimos al inglés, el vocablo *switch* hace referencia a algún dispositivo que permite abrir y cerrar un circuito, básicamente un interruptor. Pero aquí se asocia el término como conmutador. En el mundo de las redes, un switch es un dispositivo que permite distribuir datos entre diferentes computadoras pero a sólo aquellas a las que los datos van destinados. ¿Cómo es esto?

A diferencia de un hub, que distribuye los datos que le llegan entre todas las computadoras que tiene conectadas en sus puertos, un switch envía los datos sólo a aquellas máquinas que son destino de dichos datos. Para ello, cada vez que una computadora envía por primera vez un paquete de datos a través del switch, éste recibe el paquete, lee su encabezamiento y detecta un número único que identifica desde qué placa de red se originó dicho paquete. Este número se conoce como dirección MAC.

La dirección MAC (por ***M**edia **A**ccess **C**ontrol*, Control de Acceso al Medio) es un número único que identifica a cada placa de red (NIC, Network Interface Card). Dicho número es único para cada NIC que se adquiera en cualquier parte del mundo. Se lo reconoce con un número de 12 dígitos hexadecimales (48 bits), de los cuales los 6 primeros dígitos (24 bits) corresponden a la identificación universal del fabricante y los últimos 6 dígitos (24 bits) identifican al producto de dicho fabricante. Podemos ver la dirección MAC de nuestra NIC si ejecutamos el comando *ipconfig /all* desde Windows 2000 o Windows XP o el comando *winipcfg* desde Windows 98. Dicho número es almacenado en una tabla de direcciones MAC dentro del switch la primera vez que un host envía datos a través del mismo. En los siguientes envíos, el switch reconoce el host que transmite, leyendo su dirección MAC desde su tabla. De esta forma crea un registro completo de las unidades que tiene conectadas en cada puerto y permite agilizar el proceso de transmisión de paquetes asociando el par puerto-dirección MAC.
Este proceso convierte al switch en un conmutador de paquetes ya que, cada vez que llega un paquete, éste lo analiza y sabe de acuerdo a la dirección destino contenida én el mismo, a qué host enviarlo dado el registro en su tabla. Solamente envía el paquete al host destino a través del puerto correspondiente (recordemos que un hub lo envía a todos sus puertos y cada host discrimina aquellos paquetes no pertenecientes a él). Frente a un hub que divide el ancho de banda disponible para cada puerto, un switch dedica el ancho de banda disponible a cada uno de sus puertos en el momento de la conmutación, siendo ésta por turnos, dando la sensación de simultaneidad. Este proceso conlleva una notable performance en el switch frente al mero uso de hubs. Por trabajar con direcciones MAC, las cuales definen un direccionamiento físico y no lógico como en el caso de las direcciones IP, se considera al switch un dispositivo de capa 2 según el modelo de referencia OSI.

¿Por qué deberíamos utilizar un switch?

Es aconsejable pero no imprescindible hacer uso del switch. Brinda segmentación en la LAN con el consecuente aislamiento de tráfico entre segmentos. Al dedicar el ancho de banda a cada puerto en el momento de la transmisión, permite una mejora considerable en el desempeño de toda la red así como la reducción de colisiones en cada segmento.

Switch OvisLink Live-GSH8T

La empresa taiwanesa OvisLink nos ofrece un producto de excelentes prestaciones, un switch Gigabit Ethernet de 8 bocas de muy fácil implementación pero potentes características.

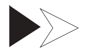
Vamos a resaltar algunas de sus características sobresalientes frente a productos similares de otros fabricantes.

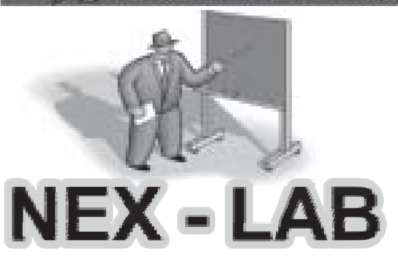
Debemos aclarar que no requiere de configuración alguna puesto que aquellos parámetros que generalmente pueden ser configurables, se autogestionan en este modelo de switch, no necesitando ingresar comandos para realizar modificaciones.

Por ese mismo motivo carece de puerto de consola RS-232. Se trata de un producto que posee tres velocidades de transmisión para acoplarse a dispositivos no tan nuevos que funcionen a tasas de transferencia más bajas. Permite trabajar a 10Mb/s, 100Mb/s e incorpora una nueva velocidad de transmisión a 1000Mb/s ó su equivalente 1Gb/s (Giga bits por segundo) del cual toma su nombre dicha tecnología.

Su función de *auto-sensing* permite detectar la velocidad de transmisión de los dispositivos acoplados a sus puertos y trabajar a velocidades distintas para cada uno de ellos.

Gracias a la función *auto MDI/MDI-X* podemos utilizar cables del tipo plano para conectar el switch a cualquier dispositivo ya que cada puerto autogestiona la necesidad de invertir o no los pares transmisor - receptor (Tx / Rx) provenientes de la conexión externa.





NEX - LAB



Figura 1

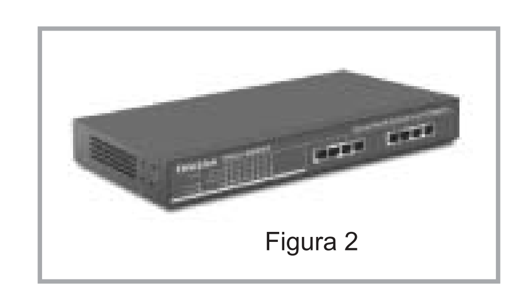


Figura 2

Con ello nos despreocupamos en pensar qué tipo de cable utilizar. Posee dos modos de transmisión en cada puerto, en sus modalidades half-duplex y full-duplex. Los puertos que funcionen a 1000 Mb/s serán forzados a trabajar en modo full-duplex. El modo de transmisión es determinado automáticamente por el dispositivo.

Su técnica de transmisión de "almacenamiento y envío" permite tener una comunicación más efectiva con la consiguiente reducción de pérdida de paquetes. ¿De qué se trata esto? En conmutación de paquetes existen dos formas de transmitirlos a través del switch: la técnica de "almacenamiento y envío" y el "método de corte". En la primera se lee completamente la trama recibida antes de enviarse, esto posee una mayor latencia debido al tiempo que tarda el switch en leer la trama completa, pero trae la ventaja de poder corregir errores mientras la va leyendo. En el segundo caso, se lee solamente la dirección destino y la trama comienza a enviarse a medida que es recibida. Es un método más rápido pero, al no ser almacenada la trama en la memoria del switch, no pueden corregirse errores y existe una mayor probabilidad de pérdida de paquetes.

Nota: Debemos aclarar que la firma OvisLink produce otro dispositivo bajo el mismo modelo de identificación, éste es Ether-GSH8T, el cual abarca y amplía las funciones del Live-GSH8T. Entre sus diferencias podemos citar que incorpora un puerto RS-232 ya que posee parámetros configurables por consola como la implementación de LAN virtuales (VLAN´s). Es una unidad que puede ser amurada ya que, originalmente, incorpora soportes para tal fin. (ver figuras 1 y 2)

Leonel F. Becchio



Un poco de historia...

Todo comenzó allá por el año 1972 en los laboratorios de la empresa Xerox Corporation´s Palo Alto Research Center (PARC). Los empleados de la firma estaban desarrollando un proyecto en conjunto y requerían interconectar sus computadoras para que trabajen junto a una impresora láser. La tarea de la interconexión le fue encomendada a Bob Metcalfe, un graduado del MIT que trabajaba en Xerox. Metcalfe tenía experiencia en el área ya que había estado trabajando sobre los desarrollos de Alohanet y Arpanet como trabajo final para su graduación. La tarea consistía, según él, en interconectar cientos de computadoras para que trabajasen velozmente junto a una impresora láser.

Sobre el final del año 1972, Metcalfe se había reunido con otros investigadores del PARC y habían logrado una red experimental cuya velocidad rondaba los 3 Mb/s. Al año siguiente, en 1973, definió los principios generales de lo que él llamaría Ethernet, primera tecnología que hizo posible formar una LAN. Ethernet es un estándar que define, entre otras cosas, el aspecto del cableado y la forma en que se transmiten los datos. Un concepto clave de Ethernet es su sistema de detección de colisiones y recuperación, llamado CSMA/CD (por Carrier Sense Multiple Access / Collision Detection). Con este protocolo, los dispositivos transmiten sólo si encuentran el canal libre y, si ambos llegasen a transmitir simultáneamente, se producen colisiones. De esta manera esperan una cantidad de tiempo aleatoria antes de proceder a retransmitir.

Recién en 1979 se logró por acuerdo utilizar 10 Mb/s como un estándar comercial. Para esa fecha, Metcalfe que ya había dejado Xerox, se había asociado con Gordon Bell de la firma Digital Equipment Corporation (DEC). Ambos propusieron a Xerox que ambas empresas se fusionen para delinear finalmente en estándar Ethernet. Para junio de ese año, Metcalfe había logrado que Intel, DEC y Xerox utilicen Ethernet como medio estándar de enviar paquetes por una red. Recién en 1983 ganó popularidad como estándar internacional al ser reconocido por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). La firma IBM intentó despojar dicho estándar creando el suyo propio, Token Ring, el cual es un mecanismo totalmente diferente, pero Ethernet fue más ampliamente utilizado. Hoy en día encontramos tecnologías Fast Ethernet (Ethernet rápido) que funcionan a 100 Mb/s y tecnologías Gigabit Ethernet a 1000 Mb/s o su equivalente 1 Gb/s.

Metcalfe denominó dicho estándar como Alto Aloha Network. Más tarde pasó a llamarse Ethernet basado en la idea del *"éter"*. El *éter* o *éter luminífero* era una sustancia hipotética a través de la cual viajaban las ondas electromagnéticas. El término fue propuesto por el filósofo griego Aristóteles y usado en varias teorías ópticas como medio de propagación de la luz, la cual se creía que era imposible que viaje en el espacio "vacío". Se supuso, en un comienzo, que el *éter* llenaba todo el Universo y era un marco estático de referencia, el cual era rígido para las ondas electromagnéticas pero completamente permeable para la materia. En 1887 los experimentos de Michelson y Morley demostraron en forma rotunda la no existencia del *éter*.

Suplemento Seguridad

Microsoft®

COR Technologies
Consultora en Capacitación Informática
Consultora en Seguridad Informática
WWW.CORTECH.COM.AR

Kerberos

Este artículo pretende dar un entendimiento de Kerberos: para qué sirve, cómo funciona. La filosofía general de Kerberos. Habiendo entendido esto, podemos leer artículos más técnicos. Está basado en el artículo de Brian Tung (The Moron´s Guide to Kerberos, www.isi.edu/~brian/security/kerberos.html). Recomendamos visitar los enlaces citados en el artículo " Dónde puedo obtener más información de Kerberos?" (en este número de NEX7) para obtener más documentación.

¿Qué es Kerberos?

Kerberos es un "servicio de autenticación" desarrollado en el MIT (Massachusetts Institute of Technology). Su propósito es el de permitir a usuarios y servicios autenticarse entre ellos en los años 80. Esto es, les permite demostrar su identidad unos a otros. El MIT desarrolló Kerberos y permitió su libre distribución con la idea de reemplazar lo que era estándar para autenticarse: "autenticación por afirmación" (authentication by assertion). Autenticación por afirmación funciona del siguiente modo: cuando un usuario corre un programa que accede un servicio en la red, el programa (llamado cliente) "afirma" al servicio que corre en representación del usuario. Y eso es todo. No necesitamos recalcarlo, esto provee un muy bajo nivel de seguridad.

La alternativa de requerir la entrada del password del usuario cada vez que se accede a un servicio de red tiene dos problemas. Primero, es demandante para el usuario y segundo, es inseguro cuando se accede a servicios en una máquina remota (el password deberá viajar sin encriptar). Claramente NO aceptable.

Por lo tanto, Kerberos fue diseñado para eliminar la necesidad de demostrar la posesión de información privada o secreta (el password) divulgando esta información Kerberos mismo. Kerberos se basa en el modelo de distribución de llaves (claves, keys) desarrollado por Needham y Schroeder. Una llave (key) se usa para encriptar y desencriptar pequeños mensajes y es ella misma típicamente una secuencia de bytes corta. Llaves (keys) dan la base para autenticación en Kerberos. Todo lo relativo a Kerberos se puede encontrar en el web site del MIT (www.mit.edu/kerberos/www).

Recordemos que una rutina de encriptación toma una llave (key) y un mensaje en texto plano (plain text) y lo devuelve como texto cifrado (cipher text). Inversamente, la rutina que desencripta toma una llave y el texto cifrado y nos devuelve (si todo funcionó bien) el texto plano original. En Kerberos, la llave para encriptar y desencriptar son idénticas.

La criptografía convencional es la que utiliza llaves idénticas (o por lo menos, fácil de obtener una de otra). Por otro lado en criptografía de llave pública (public key cryptography) hay dos llaves. Una encripta y la otra desencripta y no son derivables una de otra. En muchos casos cualquiera de las dos se puede usar para encriptar y la otra para desencriptar.

¿Qué hace Kerberos?

¿Cómo permite Kerberos autenticarnos? Del mismo modo que nos autenticamos en la vida diaria. Típicamente mostramos nuestro documento de identidad. ¿Y este qué muestra? Muestra que alguna repartición del gobierno (Policia Federal, Ministerio del interior) certifica la asociación de nuestra identidad a algún aspecto físico (foto, huella digital, altura, color ojos) que aparece representado en el documento. Al documento se lo considera imposible de copiar. Otro ejemplo de documento podría ser el carnet de conductor que incluiría el nombre, dirección, fecha de nacimiento, y quizás alguna restricción (uso de anteojos). Finalmente la identificación tiene una duración limitada que aparece como la fecha de expiración.

Notemos que la demostración de la identidad requiere además un número de cosas:

1. La tarjeta no debe estar alterada (cambio de fechas, nombre, foto).
2. Que la persona que realiza la autenticación acepta a quien la emitió como de confianza.

Kerberos trabajó básicamente de la misma manera. Es usado cuando un usuario en la red trata de hacer uso de un servicio de red y el servicio requiere asegurarse que el usuario es quien dice ser.

Para esto, el usuario presentará un "ticket" (boleto, etiqueta) que ha sido emitido por el "servicio de autenticación" (Authentication Server, AS) de Kerberos.

El servicio examina el ticket para verificar la identidad del usuario. Si todo está en regla, es aceptado.

Por tanto, este ticket debe contener información que lo ligue inequívocamente al usuario. Como el usuario y el servicio no se enfrentan cara a cara como en el caso de documentos de identidad, una foto no sirve. El ticket debe demostrar que quien lo muestra conoce algo que sólo el usuario conoce, como un password. Aun más, debe haber protecciones contra atacantes que roben el ticket y lo usen un tiempo después.

¿Qué cosas asume Kerberos?

Kerberos asume algunas cosas del entorno en el que vive. Por ejemplo, que los usuarios no harán elecciones pobres de los passwords.

Si un usuario elige una password como "password" o "nada", entonces el atacante interceptará algunos mensajes encriptados y montara un "ataque de diccionario" donde probará diferentes passwords a ver cual los desencripta. Si lo logra, el atacante podrá simular ser el usuario frente a cualquiera. En forma similar Kerberos asume que las máquinas son más o menos seguras y que sólo las conexiones a través de la red son vulnerables de ser comprometidas. En otras palabras, Kerberos asume que no hay manera para un atacante de posicionarse entre el usuario y el cliente de modo de obtener el password de esta forma.

lo logra, el atacante podrá simular ser el usuario frente a cualquiera. En forma similar Kerberos asume que las máquinas son más o menos seguras y que sólo las conexiones a través de la red son vulnerables de ser comprometidas. En otras palabras, Kerberos asume que no hay manera para un atacante de posicionarse entre el usuario y el cliente de modo de obtener el password de esta forma.

Los Detalles

Veamos más en detalle cómo funciona Kerberos.

Tanto el usuario como el servicio tienen que tener sus llaves registradas con el AS. La llave del usuario se deriva del password por él elegido. La llave del servicio se elige aleatoriamente (no hay nadie que tipee una contraseña).

Para hacer esta explicación más sencilla, imaginemos que los mensajes se escriben en un pedazo de papel (en lugar de ser digitales) y se "encriptan" encerrándolos en una caja fuerte usando una llave. En este mundo de "cajas fuerte" los llamados "principales" (usuario y servicios) son creados en el AS y se registra una copia de sus llaves en el AS.

1. El usuario envía un mensaje al AS: "*Yo, "E.L. Usuario", quisiera hablar con "Server Tal"*."
2. Cuando el AS recibe este mensaje, hace 2 copias de otra nueva llave. Esta se llama "Session key". Se usará para el intercambio directo entre el usuario y el servicio.
3. Pone una de las "Session key" en la caja 1, junto con un pedazo de papel que dice "Server Tal". Cierra la caja con la llave del usuario (recordar que AS tiene una llave de usuario y otra del servicio que fueron registradas). ¿Por qué debe estar ese pedazo de papel? Recordar que la caja es un mensaje encriptado y que la "Session Key" es una secuencia de bytes. Si la caja 1 sólo contuviese la "Session Key", el usuario no sabría decir si la respuesta vino del AS o si la decriptación fue exitosa.
4. El AS pone la otra "Session Key" en la caja 2, con un papel con el nombre "E.L. Usuario" escrito en él. Cierra la caja con la llave del servicio.
5. Devuelve ambas cajas al usuario.
6. El usuario abre la caja 1 con su llave, extrae la "Session Key" y el papel que dice "Server Tal".
7. El usuario no puede abrir la caja 2 (ya que fue cerrada con la llave del servicio). En cambio escribe en un papel la hora actual (*timestamp*) y la pone en la caja 3. Cierra la caja con la "Session Key". El envía ambas cajas (2 y 3) al servicio.
8. El servicio abre la caja 2 con su propia llave, extrae la "session key" y extrae el papel con la hora. Esto demuestra la identidad del usuario.

El *timestamp* se pone en la caja 3 para prevenir que alguien copie la caja 2 (recordar que son mensajes digitales) y la use para simular ser el usuario un tiempo después. Como los relojes no funcionan exactamente en sincronía perfecta se da un pequeño margen (5 minutos es lo típico) entre el tiempo estampado en el papel y el tiempo actual. Además, el servicio mantiene una lista de "authenticators (caja 3)" recientemente enviados de modo de asegurarse que no sean reenviados dentro de la ventana de 5 minutos.

Se preguntarán cómo hace el servicio para abrir la caja 2 cuando no hay nadie "allí" para tipear el password. Lo que sucede es que la llave del servicio no se deriva en un password. Sino que se genera aleatoriamente y es luego guardada en un archivo especial llamado "service key file".

Se asume que este archivo es seguro, de modo que nadie puede copiarlo y simular ser el usuario legítimo.

En lenguaje Kerberos, la caja 2 se llama "ticket" y la caja 3 el "authenticator". En realidad el "authenticator" contiene más información que lo que dijimos más arriba.



Suplemento Seguridad

Parte de esta información es necesaria debido al hecho que este es un mensaje electrónico (por ejemplo hay un checksum). Puede también haber una llave de encriptación en el "authenticator" para dar privacidad a las comunicaciones entre el usuario y el servicio

Autenticación del servicio

A veces, el usuario puede requerir que el servicio se autentique. Para esto el servicio toma el timestamp (sello fecha-hora) del "authenticator" (caja 3), la pone en la caja 4, junto con un papel que dice "Server Tal". Lo cierra con la "session key" y lo devuelve el usuario. (Claramente debe agregar algo al timestamp; sino devolvería la caja 3).

El Ticket Granting Server (TGS) (Servidor que otorga tickets)

Hay un problema delicado con el intercambio que hemos descrito anteriormente. Se usa cada vez que el usuario quiere contactar un servicio. Pero, notemos que realizado así se deberá entrar el password (abrir caja 1 con la llave) cada vez. La solución obvia a esto es cachear (guardar en memoria o disco) la llave derivada de la password. Pero ese cacheo es peligroso.

Con una copia de esa llave, un atacante podría simular ser el usuario en cualquier momento (hasta que se cambie el password por supuesto). Kerberos resuelve este problema introduciendo un nuevo agente, llamado "Ticket Granting Server" (TGS).

El TGS es lógicamente distinto al AS, aunque ambos pueden residir en la misma máquina. Muchas veces se les refiere colectivamente como KDC (Key Destribution Center, Centro de distribución de llaves) tal como lo llamaron originalmente Needham y Schroeder. La función del TGS es como sigue: Antes de acceder a un servicio regular, el usuario requiere un ticket para contactar al TGS, tal como si fuese otro servicio regular. Este ticket se llama el "Ticket Granting Ticket (TGT)".

Después de recibir el TGT, cada vez que un usuario desea contactar un servicio, requiere un ticket no al AS, sino al TGS.

Aún más, la respuesta está encriptada, no con la llave secreta del usuario, sino con la "session key" que proveyó el AS para usar con el TGS. Dentro de esa respuesta está la nueva "session key" para usar con el servicio regular.

El resto del intercambio ahora continúa como fue descrito anteriormente.

Es parecido a lo que sucede cuando uno visita ciertas empresas. A la entrada uno muestra su documento de modo de obtener una tarjeta de "visitante". Ahora cuando se desea entrar a diferentes secciones de la empresa, en lugar de mostrar el documento una y otra vez (que puede ser robado o se puede caer y perder) uno muestra el ID de visitante, que además es solo válido por corto tiempo. Si fuese robado, uno lo podría invalidar y obtener otro rápida y fácilmente algo que no sucedería con el documento de identidad.

Aclaremos que hay una diferencia en la analogía anterior. El ministerio del interior nos emite el documento de identidad y la empresa el ID de visitante. Estas son entidades separadas física y lógicamente. Por otro lado, en el intercambio del TGT, el AS y TGS son lógicamente distintos pero usualmente físicamente idénticos (en el mismo proceso).

La ventaja que esto nos da es que mientras los password permanecen válidos por varios meses, el TGT es bueno por un periodo corto, típicamente ocho horas.

Después de esto el TGT no puede ser usado por nadie: ni el usuario o un atacante.El termino "credentials" es usado en literatura de Kerberos en referencia al ticket y "session key" en conjunto. Sin embargo a veces encontrará los terminos "ticket cache" y "credentials cache" usados en forma indistinta.

Autenticación entre Realms (reinos)

Hasta ahora, hemos considerado el caso donde hay un solo AS y un solo TGS, que pueden o no estar en la misma máquina. Mientras que el número de solicitudes sea pequeño no existe problema.

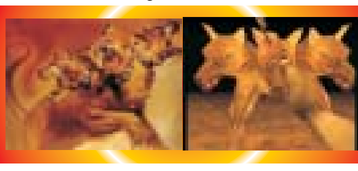
Pero cuando las redes crecen, el número de pedidos también y el AS/TGS se vuelve un cuello de botella en el proceso de autenticación.

En otras palabras este sistema "NO escalea" (usaremos esta palabra como traducción del concepto en ingles "does not scale"), y esto es muy malo para la idea de un sistema distribuido como Kerberos.

Por tanto es a veces conveniente dividir a la red en "realms" (reinos). Esta división muchas veces se hace basada en divisiones de organización de las empresas, aunque no es necesario que así sea.

Cada "realm" tiene su propio AS y TGS.

Para permitir autenticaciones entre "realms", esto es, permitir a usuarios de un "realm" acceder a servicios en otro, es necesario que el "realm" del usuario se registre en un "TGS remoto"



(RTGS, remote TGS) en el "realm" del servicio.

Notar que cuando el TGS fue agregado, un intercambio adicional fue agregado al protocolo. Aquí necesitaremos otro intercambio más: primero el usuario contacta al AS para acceder al TGS.

Luego contacta al TGS para acceder al RTGS.

Finalmente, el usuario contacta al RTGS para poder acceder al servicio deseado.

En realidad esto puede ser peor.

En el caso de existir muchos "realms" es muy ineficiente registrar cada "realm" en todos los otros. En cambio se establece una jerarquia de "realms" de modo que para acceder a un servicio en otro "realm", puede ser necesario contactar el RTGS an en el ticket.En uno o mas "realms" intermedios. Los nombres de cada "realm" se registran en el ticket.

Cómo usar Kerberos

Kerberos es normalmente una infaestructura que subyace ya preparada en los servicios (se dicen Kerberizados). Por ejemplo Windows 2000 lo utiliza en reemplazo de NTLM (LAN manager) para autenticaciones bajo Active Directory (ver nota "Kerberos ecurity en Windows 2000").

Para el usuario por tanto es transparente: él sólo entra su UID y password.Si alguien desea conocer más detalles los invitamos a leer el artículo de Brian Tung (www.isi.edu/~brian/security/kerberos.htm).

Allí podrá entender los pasos que deberá realizar un administrador unix para configurar al AS, TGS, registrar los "principals" en el AS (usuarios y servicios). Y como además deberá kerberizar los servicios (aplicaciones) de modo de utilizarlas con autenticación Kerberos.

¿Qué encriptación utiliza Kerberos?

En su versión 4 requería el uso de DES (Data Encryption Standard). En su versión 5 se indica con un identificador (que dice el tipo de cifrado a utilizar). Entonces se puede usar cualquier técnica de encriptación (cifrado).

Pregunta CompTIA

Examen Security+

¿Cuál de las siguientes situaciones puede indicar que un Hacker está atacando una Red?

A. Un ligero aumento en el tráfico de la red.
B. Un ligero descenso en el tráfico de la red
C. Un gran aumento en el tráfico del protocolo ICMP.
D. Un router (enrutador) que está transmitiendo tráfico.

Respuesta Correcta: **C**

Pruebése luego de leer el artículo Kerberos.

Pregunta: Veo las siglas TGT y TGS usados con frecuencia. Que significan?

TGT es la sigla de "Ticket Granting Ticket". TGS es la sigla de "Ticket Granting Service". Podría parecer que son intercambiables pero no lo son. Se refieren a dos conceptos diferentes dentro de la infraestructura Kerberos: El TGT es un "ticket" Kerberos para el TGS. Ambos juegan un rol especial en Kerberos. Cuando un usuario primero se autentica a Kerberos, se dirige al AS (Authentication Service) en el KDC (Key distribution Center) de modo de obtener un TGT. Este ticket está encriptado con una llave derivada del password del usuario. Cuando un usuario quiere acceder a un servicio Kerberizado, él usa el TGT para dialogar con el TGS (que también corre en el KDC). El TGS verifica la identidad del usuario usando el TGT y emite otro ticket usar el servicio deseado. La razón de porque existe el TGT es para que el usuario no tenga que entrar su password cada vez que desee acceder a un servicio Kerberizado o tener que mantener una copia de su password dando vueltas. Si el TGT es comprometido, un atacante podra utilizarlo pero sólo hasta que éste expira.

Si esta fue su respuesta entendió de qué se trata Kerberos

Kerberos security en Windows 2000

Cuando se trata de seguridad, Windows 2000 (Win2K) ofrece muchas mejoras sobre Windows NT. Probablemente, el avance más grande ha estado en el protocolo primario de autenticación del Sistema Operativo. NT LAN Manager (NTLM) ha sido el protocolo primario de autenticación para todas las versiones de NT. Win2K soporta los protocolos de autenticación NTLM y Secure Socket Layer/Transport Layer Security (SSL/TLS). Pero el protocolo primario de autenticación de Win2K es Kerberos 5, que toma su nombre de Cerberus, el perro de tres cabezas de la mitología griega que cuidaba las puertas de Hades. Echemos una mirada más de cerca a Kerberos 5 y cómo trabaja la seguridad de Kerberos en Win2K.

El protocolo de seguridad de Kerberos utiliza el concepto de clave secreta compartida para la autenticación. La encriptación se realiza con el algoritmo Data Encryption Standard (DES). Una clave secreta puede ser tanto privada (conocida por una sola parte) como compartida (conocida por dos o más partes). En Kerberos, tanto el cliente como el servicio Kerberos mantienen una contraseña encriptada, conocida como clave de sesión. El servicio de Kerberos publica el ticket (o el certificado) que contiene, entre otras cosas, el nombre de los usuarios, una clave de sesión, y la fecha de vencimiento para el ticket. Si el ticket vence durante una sesión activa, el servicio de Kerberos informa al cliente y al servidor que ha vencido el ticket y genera una nueva clave de sesión de modo que la sesión pueda continuar. El ticket inicial deja que Win2K autentique al cliente en la red de modo que el cliente pueda tener acceso a recursos en la red de la empresa (desde cualquier punto) usando un usuario y contraseña.
Kerberos consiste en varios sub-protocolos y puede funcionar a través de dominios. El centro de distribución de claves (Key Distribution Center - KDC), el servicio de autenticación de Kerberos, funciona como servicio en cada controlador de dominio. El cliente de Kerberos reside en cada cliente y servidor de Win2K. El protocolo de Kerberos consiste en tres componentes principales:
- El AS, la base de datos central que contiene toda la información sobre los usuarios
- El cliente
- El servidor

Los clientes Kerberos pueden probar su identidad al servidor sin enviar contraseñas a través de la red. Digamos que un cliente quiere acceder a una carpeta compartida en un servidor de recursos, tal como un servidor de archivo. Así es como funciona el proceso:

1. Un cliente solicita autenticación a un servidor de autenticación, tal como un controlador de dominio.
2. El servidor crea dos copias de la clave de sesión y las envía al cliente en un mensaje encriptado que con la clave privada del cliente y del servidor de recursos. La clave de la sesión contiene una marca de tiempo para prevenir que alguien capture los paquetes con un protocolo analizador y los use de nuevo para engañar al servidor.
3. El cliente envía copias de la clave de sesión al servidor de recursos, que utiliza su clave privada para abrir la clave encriptada.
4. El cliente verificado puede ahora comunicarse con el servidor porque cliente y servidor tienen la clave de la sesión secreta compartida.
5. En un paso opcional, el cliente pudo pedir a Kerberos que verifique también el servidor.

El modelo de seguridad de Kerberos de Win2K tiene varias ventajas sobre los anteriores protocolos de autenticación, incluyendo:

Relaciones de confianza transitiva para la autenticación entre dominios. Los servicio de Kerberizados aceptan las credenciales de autenticación que publique de cualquier servicio de Kerberos de un árbol de dominios, o cualquier árbol de dominio dentro de un bosque (forest). (un árbol de dominio es una organización jerárquica de dominios. Un bosque consiste en unos o más árboles de dominio.) Las relaciones de confianza son transitivas y jerárquicas; si el dominio A confía en B, y B confía en C, entonces A también confía en C.
Autenticación eficiente. Cualquier servidor de Win2K puede verificar las credenciales de los clientes sin chequear con el servicio de Kerberos en el controlador de dominio.
Autenticación mutua de cliente y servidor. Kerberos no autentica sólo el cliente; el cliente puede solicitar que Kerberos verifique también el servidor.
Autenticación delegada. Si la transacción del cliente necesita pasar a través de servidores múltiples, el servidor que verifica puede autenticar al cliente en el servidor solicitado a nombre del cliente. Usted puede instalar un número ilimitado de servidores consecutivos para delegar la autenticación.

Kerberos representa un cambio importante en la manera en que un SO autentica a clientes en una red de Windows.
La interoperabilidad del Kerberos deja que Win2K autentique a clientes no-Microsoft de otras plataformas (ej. UNIX), mientras los clientes implementan Kerberos 5. Kerberos ofrece seguridad mejorada, autenticación eficiente, y flexibilidad.
Es sorprendente lo que un perro de tres cabezas puede hacer.

Pregunta (ISC)²

Examen CISSP
¿Cuál de las siguientes siglas corresponde a un método de criptografía simétrica?

A. DES
B. RSA
C. SSL
D. SHA1

Respuesta Correcta: **A**

DES es *Data Encription Standard* (Estándar de Encriptación de Datos)
RSA es *Rivest, Shamir, Adleman*
SSL es *Secure Sockets Layer* (Capa de Conexión Segura)
SHA1 es *Secure Hash Algorithm 1* (Algoritmo de Síntesis Segura)

¿De dónde viene el nombre Kerberos?

El nombre Kerberos viene de la mitología Griega; es un perro con tres cabezas que cuidaba las puertas de Hades. (Hades era un dios pero también la morada de los muertos)

Hey! Yo recuerdo mis clases de mitología griega y el perro que cuidaba la entrada se llamaba Cerberus! ¿Cuál es correcto?

Muchos se preguntan lo anterior. Aquí van algunas respuestas:

From: Tom Yu: "Cerberus" es como se escribe en latín la palabra Griega "Kerberos", y según el OED (Oxford English Dictionary) se pronuncia "serberus", que está en desacuerdo con el griego donde la consonante inicial es una "k". El proyecto Athenas del MIT eligió usar la escritura y pronunciación Griega.

From: Jan Sacharuk: Tom Yu tiene razón, Cerberus es en latín. Sin embargo, el hecho que el OED dice que "c" se pronuncia como "s" es una afección inglesa. En latín, la letra "c" es siempre dura. Así que Cerberus es pronunciado "Ker-ber-ous". La pronunciación de la letra "u" es también un poco diferente, estando entre "us" y "ous".



From: Michael A. Covington: "Kerberos" es el nombre Griego original. En Latín, la letra K no se usa normalmente, y en tiempos Romanos, la C siempre represento el sonido K. También, "-os" es un sufijo Griego (nominativo masculino singular) cuyo equivalente más cercano en latín es el sufijo "-us" (muy común en nombres en Latín). Por eso el nombre pasa al latín como Cerberus. (¡Ven, un Doctorado en Lingüística es útil para algo!)



Diseñando un sistema de autenticación: un diálogo en 4 escenas

A continuación mostramos el abstract del trabajo "diseñando un sistema de autenticación: un diálogo en 4 escenas". Es un artículo introductorio a Kerberos, presentado como una obra de teatro en cuatro escenas. Brillante. Lo recomendamos para quien desee comprender Kerberos de un modo divertido.

Este diálogo provee una narración ficticia del diseño de un sistema de autenticación llamado "Charon". A medida que progresa el diálogo, los personajes Athena y Eurípides descubren los problemas de seguridad inherentes a un entorno de redes abierto. Cada problema debe ser abordado en el diseño de Chaon, y el diseño evoluciona de acuerdo a esto. Athena y Eurípides no completan su trabajo hasta que el diálogo se cierra.

Cuando terminan el diseño del sistema, Athena cambia el nombre del sistema a"Kerberos". Este nombre coincide por mera casualidad con el sistema de autenticación que fue diseñado e implementado en el MIT bajo el nombre de proyecto"Athena". El sistema "Kerberos" de este diálogo guarda gran similitud con el sistema descrito en *Kerberos: An Authentication Service for Open Network Systems* presentado en el Winter USENIX 1988, en Dallas, Texas.

Se puede encontrar en:
http://web.mit.edu/kerberos/www/dialogue.html



¿Dónde puedo obtener más información de Kerberos?

Si Ud es Nuevo en Kerberos recomendamos:

-Bill Bryant, "Designing an Authentication System: A Dialogue in Four Scenes." http://web.mit.edu/kerberos/www/dialogue.html

Una explicación simple del protocolo Kerberos. El lenguaje técnico se mantiene a un mínimo.

-Jeffrey I. Schiller, "Secure Distributed Computing", Scientific American, November 1994, pp 72-76.

Un pantallazo excelente que cubre todos los detalles importantes del protocolo Kerberos. También explica cómo es usado en el MIT como un ejemplo de la "vida real".

-J. G. Steiner, B. Clifford Neuman, and J.I. Schiller, "Kerberos: An Authentication Service for Open Network Systems". ftp://athenadist.mit.edu/pub/kerberos/doc/usenix.PS

Este es el trabajo original donde se describe Kerberos. Es un review general muy bueno. Describe la notación de encriptación que es usada en muchos otros trabajos sobre Kerberos. Así que vale la pena leer para aquel que quiere ahondar en la literatura.

-Brian Tung, "The Moron's Guide to Kerberos" http://www.isi.edu/~brian/security/kerberos.html. A pesar de su título, va bastante en detalles. Este es el trabajo sobre el que basamos nuestro artículo del suplemento de seguridad de NEX7.

-La página web de Kerberos de MIT: (http://web.mit.edu/kerberos/www/) tiene muchos links que apuntan a recursos sobre Kerberos.

Uno de los mejores tutoriales para Kerberos es el artículo de Jim Rowe, "How To Kerberize Your Site" (Cómo Kerberizar su sitio), que está disponible en http://www.y12.doe.gov/~jar/HowToKerb.html

Hay también un RFC de Kerberos 5: RFC 1510, disponible en:

http://www.ietf.org/rfc/rfc1510.txt

Difícil de leer a menos que ya sepamos algo de como funciona Kerberos.

¿Es conveniente certificarse en Linux?

¿Qué certificación me conviene tener en el mercado argentino?

Mucha gente está muy capacitada en Linux, pero muy pocos tienen certificaciones internacionales. Por ejemplo hasta hace poco (mediados del 2003) había solo 4 personas en Argentina certificados en LPI nivel 2. La primera pregunta es si vale la pena certificarse en Linux. La respuesta es sí, ya que aquél que realiza el esfuerzo (tanto de aprendizaje como económico) mostrará a un futuro empleador su decisión de educarse y búsqueda de logros. Además, al igual que en la obtención de un titulo universitario o en la realización de una tesis, nos permite englobar mucho de los conocimientos esparcidos que poseemos. A pesar de ser muy expertos es posible que ciertos temas no hayan pasado por nuestro aprendizaje. Por lo tanto tener que cubrir una currícula completa nos obliga a llenar nuestras falencias. A la hora de ser promovidos o seleccionados en un nuevo empleo el haber realizado dicha certificación nos distinguirá. Lamentablemente los costos (U$S 100.- por examen para LPI y varios cientos U$S en el caso Red Hat) son muy altos para el mercado argentino con relación a los salarios actuales.

La segunda pregunta es a qué certificación apunto. Existen varias: Red Hat, LPI, SUSE, Novell y otras. En la Argentina las más populares son LPI y Red Hat. La certificación Novell es muy reciente (recordar que Novell adquirió a Ximian y Suse hace relativamente poco). LPIC es posible rendirla utilizando empresas como VUE (Virtual University Enterprise, www.vue.com) y Prometric (www.prometric.com). Red Hat debe realizarse en Red Hat mismo. En Argentina Preteco SA (www.preteco.com) viene realizando cursos y exámenes a precios promocionales y los exámenes son rendidos en Argentina.

Pero volvamos a la pregunta de cuál elijo. Tanto la certificación Red Hat como la LPIC son respetadas y la respuesta dependerá de 2 factores:

1. inversión que quiero hacer
2. El mercado al cual apunte (recordar que Red Hat Enterprise es ahora una distribución que apunta al mercado corporativo).

Ambas nos forman en Linux (Unix-like systems) y garantizan que poseemos un conocimiento avanzado de Linux.

Muchas veces se nos consulta sobre qué debo estudiar: expertise en Linux o Microsoft. La respuesta creo viene dada en el párrafo que sigue:

Es IMPOSIBLE NO conseguir hoy (Marzo 2004) un excelente empleo en Argentina para alguien con una certificación LPIC nivel 2 o RHCE, complementada con la certificación de Microsoft MCSE (especialidad seguridad) y con conocimientos avanzados en seguridad informática.

Si esa misma persona no tuviese las certificaciones (pero los mismos conocimientos) habría que reemplazar "IMPOSIBLE NO" del párrafo anterior por "PROBABLE".

Pregunta Microsoft

Examen 70-296 - Planning, Implementing and Maintaining a Microsoft Windows Server 2003 Environment for an MCSE 2000

Usted es el administrador de redes de su compañía. La red corporativa consiste de un dominio de Active Directory donde todos los servidores corren bajo Windows Server 2003 y todos los clientes bajo Windows XP Professional. Está planeando instalar Software Update Service (SUS) en un servidor llamado SUS1 y quiere testear las actualizaciones en computadoras específicas y aprobar las actualizaciones apropiadas antes de implementarlas en la red.

¿Cuál de las siguientes tareas debería realizar?

a. En un GPO (Group Policy Object = Objeto - Política de Grupo), especificar SUS1 com ubicación del servicio de actualización y aplicar el GPO a las computadoras.
b. En un GPO, especificar SUS1 como ubicación del servicio de actualización y aplicar el GPO a los usuarios.
c. En un GPO, especificar que las actualizaciones se bajarán de SUS1 y aplicar el GPO a las computadoras.
d. En un GPO, especificar que las actualizaciones se bajarán de SUS1 y aplicar el GPO a los usuarios.

Respuesta Correcta: **A**

Puede usar SUS para proveer sólo actualizaciones testeadas a las computadoras en su red corporativa. Puede configurar un servidor SUS para descargar actualizaciones del sitio de actualizaciones de Microsoft, de otro servidor SUS o de un punto específico de distribución de contenido, como carpetas compartidas configuradas manualmente en un servidor de archivos. Como alternativa, puede configurar el servidor SUS para descargar sólo la "metadata" asociada a las actualizaciones disponibles. Puede testear las actualizaciones disponibles y aprobar sólo las que usted considere apropiadas. Las computadoras cliente con *Automatic Update* se conectarán al servidor SUS especificado y descargarán la lista de actualizaciones aprobadas. Si usted configura el servidor SUS para almacenar los archivos de actualizaciones vigentes localmente, entonces las computadoras cliente lo descargarán del servidor SUS. Si configura el servidor SUS para no almacenar los archivos de actualizaciones vigentes, entonces las computadoras cliente los descargarán del sitio de Microsoft.

En este escenario, usted debería configurar **SUS1** para descargar las actualizaciones del sitio de Microsoft y almacenarlos localmente. En un GPO, debería especificar **SUS1** en la política *"Especificar la ubicación del servicio de actualizaciones de Microsoft para Intranet"* (*Specify Intranet Microsoft Update Service Location*). Esta es una política para una computadora específica, por lo tanto, debería aplicar el GPO a computadoras conectándola a una unidad organizacional (OU), donde residen todas las computadoras apropiadas. No puede especificar que las actualizaciones se bajarán de SUS1 en un GPO, ésta es una opción SUS que puede ser configurada sólo en las **Opciones de Configuración** de la página Web de administración en cada servidor SUS.

Preguntas LPI (Linux Professional Institute)

Examen 117-101-RPM

Luego de ejecutar el comando

`crontab -e`

¿Cuál de las siguientes líneas debe agregar a **crontab** para que el script `prueba.sh` se ejecute el primer día de cada mes a las 2:05 de la madrugada?

A. `2 5 * 1 * prueba.sh`
B. `5 2 1 1 * prueba.sh`
C. `5 2 1 * * prueba.sh`
D. `2 5 * * 1 prueba.sh`

Respuesta correcta: **C**.

`crontab` interpreta cada línea con la siguiente sintaxis:

<minuto> <hora> <dia_del_mes> <mes> <dia_de_semana> comando

donde el * significa cualquier valor posible.

Examen 117-201

¿Cuál de los siguientes comandos protege un archivo llamado `prueba.txt` de ser borrado por `root`?

A. `chattr +c prueba.txt`
B. `chattr +i prueba.txt`
C. `chmod 000 prueba.txt`
D. `chmod a-rwx prueba.txt`

Respuesta correcta: **B**.

El modificador `i` de `chattr` se refiere a la propiedad de inmutabilidad de un archivo. Cuando la inmutabilidad de un archivo está establecida ningún usuario (ni siquiera `root`) puede borrar, modificar o renombrar un archivo. Sólo el usuario `root` puede establecer (con `chattr +i archivo`) o quitar (con `chattr -i archivo`) la propiedad de inmutabilidad a un archivo.

TODOS LOS CAMINOS LLEVAN A DEBIAN

¿Qué es Debian? ¿Por qué Debian?



¿Que es Debian?

Debian es una asociación de personas con un fin común, un proyecto iniciado hace más de 10 años orientado al desarrollo de un sistema operativo libre. A que llamamos sistema operativo? A un conjunto de programas y utilidades básicas que brindan el funcionamiento de una computadora. El centro de un SO es el kernel. En la actualidad Debian usa para su sistema operativo el kernel de Linux. Sin embargo se está trabajando para ofrecer a Debian con otros kernels, concretamente el **núcleo Hurd**.

Obviamente lo que la gente en su mayoría quiere es el software de aplicación, herramientas que lo ayuden desde editar un documento a navegar en Internet. Debian incluye la mayor cantidad de software disponible para una distribución. Desde su sitio oficial (www.debian.org) podemos descargar en su versión estable con 8710 **paquetes deb** los 8 isos de la última **versión oficial** 3.0r2 **codename woody**.

Siendo en sus comienzos la distribución oficial de la **Free Software Foundation** (www.fsf.org) en la actualidad mantiene una estrecha relación con la FSF por ser la distribución que respeta al pie de la letra el proyecto **GNU** (www.gnu.org). Por todo esto, y accediendo el pedido de la FSF, la distribución de Linux de Debian lleva el nombre **Debian GNU/Linux** y no solo Debian Linux, por usar las herramientas GNU y el kernel de Linux.

¿Qué ventajas tiene Debian?

Totalmente actualizable: Debian es la única distribución que puede decir abiertamente que desde la versión 2.1 hasta la última el sistema se puede actualizar total y completamente sin perder ninguna de las funcionalidades actuales. Esto, en otras distribuciones en impracticable. Esto se debe principalmente al cuidado y control que se tiene sobre las aplicaciones que se desarrollan, el sistema de **paquetes deb** y la aplicación que administra estos paquetes: **el apt-get**. En cualquier otra distribución, actualizar una versión a otra es impracticable, la única posibilidad es guardar los datos actuales e instalar de cero para que todo quede realmente bien.

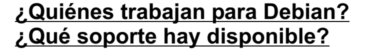
Arquitecturas que soporta: La distribución completa se encuentra portada a la mayoría de las arquitecturas conocidas, entre ellas todas las variantes de PC (Intel, AMD, Cyrix, Via), SUN SPARC, MK68K (Atari, Amiga y Apple), ALPHA, POWERPC, MIPS (Cobalt), ARM, HPPA (HP pa-risc), IA-64 (Intel 64bits), S390 (IBM S/390 series). Por otro lado, está casi terminado el port a AMD64 para soportar los microprocesadores Opteron tanto en 32bits como en 64bits. A su vez, el kernel de Linux no es el único disponibles, también esta en desarrollo **el kernel Hurd** que es un desarrollo propio y libre de la FSF y el kernel de FreeBSD y NetBSD.

Debian es libre: Hay un **contrato social** el cual se respeta a la hora de tomar decisiones. En muy resumidas cuentas, el contrato social asegura que Debian continuará siendo 100% libre por siempre, todo el software desarrollado estará disponible para la comunidad, no se ocultarán problemas y que todos los programas en la distribución tienen que cumplir el standard de ser libres. Si bien existe la posibilidad de tener software que no cumpla este último requerimiento estarán disponibles en otras secciones llamadas **contrib** y **non-free** y no tienen que ser herramientas fundamentales para el sistema.

No existen los problemas de dependencias: Debian utiliza el formato deb y para instalar una aplicación sólo alcanza con un comando en consola podemos actualizar todo el equipo, paquetes concretos, instalar y desinstalar aplicaciones. El **apt-get** simplifica la administración de paquetes y nos hace olvidar de las dependencias que tantos problemas traen en distribuciones rpm.

Sencillamente el sistema les avisará que es lo que va a necesitar instalar para satisfacer las dependencias de la aplicación y procederá a conseguirlas.

Si no tienen los CDS, sencillamente se conectará al mirror de Debian más cercano y se bajará lo que necesite. Todo totalmente compatible con lo que ya había instalado manteniendo una base de datos de lo instalado y lo disponible para instalar en los mirrors y de esta forma controlar de tener la versión correcta.



Se respeta el Filesystem Hierarchy Standard: El estándar determina la ubicación en el árbol de directorio de las aplicaciones y archivos de configuración y este estándar es respetado totalmente. Todos los archivos de configuración están en /etc. Las aplicaciones esenciales del administrador en /sbin, etc. De esta forma se garantiza la compatibilidad con otras distribuciones y otros sistemas UNIX.

Fuentes: Para toda aplicación existen sus fuentes y la posibilidad de instalarlas con la misma simpleza que con el "apt-get install aplicación".

También se pueden conseguir las fuentes e instalarlas personalizándolas a nuestro gusto y aprovechando nuestro hardware como hace gentoo (www.gentoo.org) con su sistema de ports basado en Free BSD.

Calidad y testeo: El proyecto le presta más atención a la calidad y al testeo que al lanzamiento de nuevas versiones para que la gente las pruebe en sus equipos.

Por esto es que no hay lanzamientos seguido dado que para que haya que generar una versión nueva de la distribución tiene que haber pasado un período en el cual se hayan testeado todas y cada una de las aplicaciones y no haya ningún bug reportado. Esto lleva a tener una distribución robusta y probada sin problemas ni agujeros de seguridad. Se utiliza el **Bug Tracking System** como base de datos para tener un control de los problemas reportados y su status. La mayoría de los problemas son solucionados en días y todos están publicados en la página.

para tener un control de los problemas reportados y su status. La mayoría de los problemas son solucionados en días y todos están publicados en la página.

Varias versiones en simultáneo: A diferencia de otras distribuciones, para que una aplicación sea aceptada en la distribución tiene que haber cumplido cierto periodo de madurez, aparte de respetar los standars. Para esto, existen tres versiones en simultaneo, **stable, testing y unstable**.

La versión **stable** es la que se encuentra en la versión 3.0r2 codename woody y contiene los 8710 paquetes. En stable Todas las aplicaciones están totalmente probadas. Por otro lado esta la versión **testing (sarge)** que tiene muchas mas aplicaciones en sus versiones mas recientes. Una aplicación se encuentra en testing porque depende de alguna aplicación que esta todavía en **unstable (sid).** Las aplicaciones que se encuentran en **unstable** no están completamente probadas, tienen errores sin resolver y son versiones más actuales que las de testing y stable.

En **unstable** hay más de 12000 aplicaciones. Si bien las aplicaciones en unstable tiene errores o no están totalmente probadas, son igual de estables que cualquier distribución del mercado que anda por su versión 9 y personalmente uso la versión unstable hace tiempo para mi workstation sin mayores inconvenientes.

Disponible en varios idiomas: No es una virtud exclusiva de Debian, cualquier distribución podría tener sus programas y manpages en varios idiomas. Pero Debian se encarga de mantener su distribución en los 13 idiomas mas usados, inclusive las manpages.

¿Quiénes trabajan para Debian? ¿Qué soporte hay disponible?

Con más de 1200 desarrolladores, respetando el **contrato social** de Debian, es la distribución con más gente colaborando.

Cuando hablamos de colaboradores hacemos hincapié en el hecho de que Debian es un proyecto sin fines de lucro a diferencia de las "distribuciones .com". Una de las ventajas de esto es que Debian jamás cancelara el presupuesto para el desarrollo de su distribución como Corel Linux, nunca discontinuara la distribución libre de su distribución para fomentar la venta de su versión enterprise como RedHat Linux, ni tampoco existe la posibilidad de que se tenga que presentar en quiebra como Mandrake Linux.

Sin embargo el no tener un soporte económico tiene varias desventajas, a saber:

*Al no existir la posibilidad de cerrar un trato económico empresas como Oracle y Borland no certifican a Debian como una distribución soportada por ellos.

*No existe soporte oficial, ni siquiera durante o meses después de la instalación como si tienen las distribuciones .com.

Si bien esto en primera instancia aparenta ser un gran problema a la hora de introducir a Debian en una empresa, tanto Oracle como Kylix (el entorno de desarrollo de Delphi para Linux) funcionan sin inconvenientes. Mientras que el Kylix estando certificado para RedHat tiene problemas en la instalación y sólo funciona con versiones concretas de RedHat.

En cuanto al soporte oficial, Debian es conocida por ser la distribución con mayor documentación disponible y por respetar todos y cada uno de los estandáres. Cualquier empresa que brinde soporte para Linux tiene la capacidad de dar soporte para Debian.

Algunas otras quejas comunes con Debian GNU/Linux.

El **dselect** es confuso para aprender: El dselect es el frontend para el management de paquetes en Debian. En la actualidad ya esta superado y reemplazado por otros frontends como el **aptitude** para consola o el **synaptic** para modo gráfico. El problema concreto con el dselect es que cuando fue concebido no había semejante cantidad de paquetes disponibles y crea mucha confusión tener que browsear 12mil paquetes en una aplicación como el dselect.

Falta **software comercial popular:** Es sabido que mucho software comercial no esta disponible para Linux. Toda la suite de **Macromedia**, a excepción del reproductor para flash, de momento no esta disponible.

Hay un proyecto concreto de **Macromedia** para portar sus aplicaciones a Linux. La falta de aplicaciones como el Word o el Excel ya no debería existir a menos que hagamos cosas muy elaboradas con el excel. Hay suites de oficina completas como el OpenOffice, el Koffice o el GnomeOffice.

Para los que trabajen con bases de datos, Debian ofrece tanto mySQL como postgreSQL y soporta SAP DB, Oracle, Informix, IBM DB2 y otros.

Muchos otros productos comerciales están siendo desarrollados a medida que las empresas se dan cuenta del mercado que pierden por no introducir sus productos en Linux.

Por ser libremente distribuible es difícil calcular con exactitud la cantidad de usuarios de Linux, pero se estima que unos 15 millones de usuarios tienen Linux como sistema operativo.

Linux es difícil de configurar: Esta es un área en la que se esta trabajando activamente, mucha gente considera que Linux es mas sencillo de instalar que Windows. Hardware como las impresoras, tienen configuraciones muy sencillas y llevables para alguien que no tiene experiencia previa.No todo el hardware esta soportado por Linux: Hardware muy particular, muy raro, muy viejo o muy nuevo pueden llegar a tener problemas.

Concretamente son aquellos

en los que el fabricante sólo desarrolló el driver para Windows (modems por ejemplo). Igualmente en la mayoría de los casos hardware equivalente está disponible para Linux. Algunos fabricantes directamente optan por no detallar las especificaciones técnicas del producto como para poder darle soporte. Esta es una área en la que se está trabajando.

¿Quiénes usan Debian GNU/Linux? ¿Hay otras distribuciones basadas en Debian?

Luego de RedHat, es la distribución más utilizada en sitios de Internet, sólo en nuestro país es utilizado en las siguientes entidades:

Sitio oficial Presidencia de la Nación Argentina
http://www.presidencia.gov.ar
Secretaria de Medios de la Nación
www.medios.gov.ar
Universidad Nacional de Buenos Aires
www.uba.ar
Gobierno de la Ciudad de Buenos Aires
www.buenosaires.gov.ar
Movicom: www.movicom.com.ar
LUGar: www.linux.org.ar

Por otro lado, el ministerio de educación utiliza un proyecto nacional llamado Ututo (www.ututo.org), una **distribución basada en Debian y Suse** para brindarles conectividad a las escuelas. De hecho hay varias distribuciones basadas en Debian. Desde la muy conocida **Knoppix** y Xandros hasta la discontinuada Corel Linux. Sólo por mencionar algunas podemos nombrar a: Adamantix, DemoLinux, Gibraltar, Libranet, Lindows, Linex, Morphix, Progeny, Stormix, etc.

¿Debian es gratis?

Uno puede preguntarse: ¿por qué gastará la gente horas de su propio tiempo escribiendo software, empaquetándolo cuidadosamente, y luego regalándolo? Las respuestas son tan variadas como la gente que contribuye. A algunas personas les gusta ayudar a otras. Muchas escriben programas para aprender más acerca de las computadoras. Más y más personas están buscando la manera de evitar los precios inflados del software. Un grupo creciente contribuye como un agradecimiento por todo el excelente software libre y gratuito que ha recibido de otros. En las instituciones académicas muchos crean software libre para ayudar a obtener los resultados de sus investigaciones en un uso más amplio. Las empresas ayudan a mantener el software libre para poder observar cómo se desarrolla éste, ¡no hay una manera más rápida de obtener una nueva característica que implementarla uno mismo! Desde luego, muchos de nosotros sólo lo encontramos divertido.

Debian está tan comprometido con el software libre que este compromiso se encuentra formalizado en el **Contrato Social**. Aunque Debian cree en el software libre, existen casos en los que la gente quiere o necesita disponer de software que no es libre en sus máquinas. Siempre que sea posible, Debian respaldará esto.

Hay un número creciente de paquetes que tienen como única misión instalar software que no es libre en un sistema Debian. Como por ejemplo el caso de las fuentes truetype de Microsoft. El paquete Debian sólo baja las fuentes libremente como cualquier otra persona del sitio de Microsoft y las instala. Sólo tenemos disponible para bajar del mirror Debian la herramienta para luego bajar las fuentes.

¿Gratis? ¡Pero los CDs y el ancho de banda cuestan dinero!

Uno podría estar preguntándose: si el software es gratuito, ¿por qué tengo que pagar a un vendedor por un CD, o pagar a un ISP (un proveedor de Internet) para descargarlo? Cuando compra un CD, está pagando por el tiempo de alguien, por la inversión hecha para crear los CDs , y el riesgo (en caso de que no los pueda vender todos). En otras palabras, está pagando por el medio físico usado para distribuir el software, no por el software en sí.

Cuando usamos el término «libre», nos referimos a la libertad del software, no a que tenga que ser gratuito (en inglés, «free» es sinónimo de «gratuito» y de «libre»). Puede leer más sobre en el sitio de la Free Software Foundation (www.fsf.org).

Hay algo que vale aclarar: es muy común cuando la gente va al sitio de Debian a bajarse los CDs, sorprenderse con que estén disponibles o en DVD o en CD (8 isos). **No hace falta bajar todos los CDS,** con tan solo tener el CD 1 (de 650mb) o el CD de instalación por red (50mb aprox.) o alguno de los disquetes de instalación por red uno puede instalar lo que necesite via Internet y solo bajarse lo que va a usar.

La mayoría del software cuesta cientos de dólares. ¿Cómo es que Debian lo regala?

Una pregunta mejor es, ¿cómo hacen las compañías de software para venderlo a un precio tan alto? Crear software no es como hacer coches. **Una vez que se ha conseguido una copia del software, los costos de producción por hacer un millón más son mínimos**. Mírelo de ésta otra forma: si usted tiene un surtido interminable de arena en su patio trasero, podría estar deseando regalar arena. Sin embargo, sería tonto pagar un camión para que se la lleve a otras personas. Usted los haría venir y que se la lleven ellos mismos (equivalente a obtenerla de la red) o ellos podrían pagar a alguien para que se la llevara hasta su puerta (equivalente a comprar un CD). Así es exactamente cómo opera Debian y el porqué de que la mayoría de los CDs sean tan económicos.

Debian no gana ningún dinero de la venta de CDs. A la vez, se necesita dinero para pagar gastos como el registro del dominio y el hardware. Por ello, le pedimos que compre a uno de los fabricantes de CDs que donan parte del dinero a Debian.

El apt-get:

Hay varias herramientas para administrar los paquetes en Debian, todas trabajan y comparten la información. Usar una herramienta o la otra no nos va a impedir cambiar de administrador. Son diferentes formas de instalar una aplicación y más que nada son gustos. La herramienta inicial de Debian era el dpkg, ahora es el apt-get. Tanto el dpkg como el apt-get trabajan por consola. Hay frontends o GUIs para trabajar tanto en consola como en modo gráfico, entre **ellos el ya antiguo** dselect, el aptitude (muy recomendado) o en modo gráfico, el synaptic.

¿Cómo se instala y se desinstala una aplicación?

Para instalar una aplicación, está la forma sencilla, en la cual sabemos el nombre del paquete **`apt-get install paquete`** y existe otra forma, sin saber el nombre del paquete. En este caso necesitamos averiguarlo. Tenemos dos opciones: recurrir a packages.debian.org y buscar el paquete en los buscadores del sitio o usar el **`apt-cache search nombre_de_la_aplicación`**

Esto nos debería devolver nombres de paquetes. Luego sigue igual que la primera opción, **`apt-get install paquete`**.

A diferencia de una distribución RPM el apt-get calcula las dependencias y las resuelve automáticamente. Por otro lado nos ofrece software que estaría relacionado con el que queremos instalar y que seria prudente instalar. Hace falta aclarar que el apt-get sabe donde está el paquete. En otras palabras, no hace falta que tengamos que bajar nada de Internet ni tener los CDs, el apt-get sabe en qué mirror están los paquetes disponibles, los baja solo y los instala. Luego corre un mini wizard para configurarlo básicamente y tenerlo andando. ¿Cómo es este proceso en una distribución RPM?Bueno, en una distribución rpm tenemos que, primero, conseguir el paquete. Para esto podemos incursionar en **`rpmfind.net`** a ver si encontramos algo que funcione, o buscar algo desactualizado en los CDs de instalación. Luego, quedaría hacer un **`rpm -ivh paquete`**. ¿Pero qué pasa? Este paquete necesita de otro paquete y el rpm no sabe dónde está. Así que volvemos a **`rpmfind.net`** o a buscar entre nuestros CDs a ver si encontramos ese paquete. ¡Lo conseguimos! Cuando vamos a instalarlo nos pide una librería que no tenemos, bueno... nuevamente a buscar la librería a **`rpmfind.net`** o a nuestro CD. Imagínense que esto es de nunca acabar.

Finalmente, cómo se desinstala algo? Fácil: **`apt-get remove paquete`**.

¿Cómo se actualiza el sistema?

Como ya mencionamos, el apt-get mantiene una base de datos de los programas instalados y de los programas disponibles para instalar, con sus versiones. También comentamos que hay tres versiones en paralelo de Debian GNU/Linux.

Bueno, imaginemos que queremos actualizar nuestros paquetes instalados a la versión nueva que hay disponible en los mirrors de Debian.

Alcanza con hacer un:

`apt-get update && apt-get upgrade`

y sólo contestar **`yes`** cuando se nos pregunte si queremos actualizar el equipo.

El apt-get se va a encargar de bajar las aplicaciones, actualizar los paquetes, preguntarnos si queremos guardar las configuraciones nuevas o reemplazarlas y de bajar y subir los servicios actualizados para que apliquen los cambios.

Entonces con Debian, ¿cómo haría para instalar el kde?

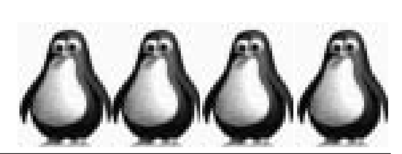
Para instalar el **kde** alcanza con hacer **`apt-get install kde`**. Nada más, el apt-get se va a encargar de todo. Esto en Debian es estándar con cualquier aplicación. **Los problemas de dependencias no existen en la versión estable.**

¿Cómo se configura el apt-get?

Para que el apt-get consiga las aplicaciones tenemos que definir dónde puede conseguirlas. El principal archivo de configuración para las fuentes de instalación es el archivo **`/etc/apt/sources.list`**. El apt-get tiene un frontend para su configuración, es el **`apt-setup`**. El **`apt-setup`** nos va a preguntar de dónde el apt-get va a conseguir sus paquetes. Podemos poner múltiples opciones, como cdrom, ftpserver, mirror de Debian, directorio, etc. Luego de haber definido todo, nos quedaría pendiente actualizar la base de datos, dado que agregamos más posibilidades desde dónde actualizar. Para esto utilizamos **`apt-get update`**. Esto nos actualiza la base de datos. No estaría mal hacer luego un **`apt-get upgrade`** porque como agregamos nuevos medios de instalación, es probable que alguno de estos medios tenga cosas nuevas en relación a lo que tenemos instalado.

¿Qué diferencias hay entre un RPM y un DEB?

Hay varias cosas para tener presentes a la hora de elegir una distribución. Una de las más notables es el sistema de paquetes. Técnicamente hablando, un paquete RPM "necesita" otro paquete como dependencia. Un paquete DEB aparte de "necesitar" un paquete también puede "sugerir" y "recomendar" un paquete. Por ejemplo, cuando instalemos el Apache se nos va a sugerir también instalar la documentación del Apache. Esto significa que cuando instalemos algo nos van a aparecer paquetes relacionados con el seleccionado que probablemente uno desee instalar.Otra ventaja es el concepto de alternativas. Un ejemplo, hay muchos clones del vi. Cuando uno escribe "vi" uno de ellos debería ejecutarse. Si tenemos más de uno instalado, cuando escribamos "vi" ¿cuál debería arrancar? En RedHat no hay un estándar para especificarlo. Algunos clones instalan el binario "vi", otros links simbólicos, todo pisando al previo "vi". En Debian el desorden no está aceptado. /usr/bin/vi es un link simbólico a /etc/alternatives/vi y ese es el link simbólico al binario. Cuando instalemos una alternativa del "vi", va a redirigir el /etc/alternatives/vi. Cuando lo desinstalemos va a redirigirlo a otro clon del "vi" que tengamos o que hayamos instalado antes. El mismo mecanismo se aplica para las manpages. "man vi" nos va a llevar a la documentación del clon del "vi" que estemos usando.





En distribuciones RPM para mantenerse en carrera ponen paquetes que ni siquiera tienen quince días de haber sido liberados en la distribución. Con lo cual su concepto de producción de distribuciones es "¡Que los usuarios sean los betatesters, si se quejan les decimos que están recibiendo software libre!" Concepto muy válido, pero para mí, que debo responder de la estabilidad de servidores, nada convincente.

Vamos a ver un caso práctico. En una distribución RPM uno tiene disponibles los paquetes que estén en el CD de instalación. Estos RPM son creados por empleados de la distribución e incluyen todo el software importante. Este CD aparte incluye mucho software y si hacemos todo nuestro trabajo en aplicaciones de ese CD probablemente no necesitemos nada más. El problema comienza cuando queramos usar otro administrador de ventanas, por ejemplo, que no este incluído en el CD. Hay que conseguirlo por separado. Para esto hay mirrors de RedHat donde conseguir software RPM, estos sitios se llaman sitios "contrib" y es donde la gente contribuye con RPM creados por ellos de sus programas favoritos.

Hay varios problemas con esto:

1. Nuevas versiones de los programas que vienen con el CD son lanzados rara vez. Por ejemplo, si uno quiere usar la última versión de un compilador hay que esperar al próximo release de la distribución.
2. Si uno instala las últimas versiones disponibles de las utilidades básicas de los mirrors de contribuciones es SEGURO que van a tener dependencias rotas y cosas que no funcionen. Gente externa a la distribución no sabe armar RPMs como sabe el personal de la distribución.
3. Programas que no vengan con la distribución hay que bajarlos por separado.

Vamos a explayarnos un poco más sobre el punto 2. El software que hayamos bajado de contrib suele no funcionar. Pero esto no quiere decir que todos los RPMs que bajemos de ahí no van a funcionar, sino que la gente que crea esos paquetes son gente que no se dedica a eso, probablemente no tenga experiencia previa y no tienen ninguna obligación de hacerlos bien. También van a notar que la misma aplicación probablemente este empaquetada dos veces con diferentes nombres. Si instalamos una versión creada por una persona y luego la actualizamos por la de otra persona, está confirmado que vamos a tener problemas.

En otras palabras, el desorden prevalece.

El modelo organizativo de Debian es totalmente diferente. Dado que Debian es una organización voluntaria, no hay límite en cuanto a desarrolladores, porque no se les paga. Entonces no hay un mirror "contrib" como en otras distribuciones con el mismo sentido que tienen las RPMs.

En Debian, cualquiera que empaquete software es un desarrollador y sus paquetes se introducen en la distribución. Ser un desarrollador para Debian no es crear un paquetes DEB y subirlo a Debian como en RedHat contrib.

Uno tiene que registrarse, ser aprobado y luego de serlo uno se hace cargo de un paquete. Esto significa que nadie más puede mantener ese paquete, uno es responsable de crear el paquete para este programa y dar soporte. Este paquete entonces pasa a formar parte de la distribución.

Cuando uno crea un paquete tiene que seguir unos pasos estrictos, que garantizan que los paquetes creados funcionen con el resto de la distribución.

El sistema tiene muchas ventajas, volvamos a instalar un paquete.

1.Como los desarrolladores están encargados de mantener sus paquetes y siempre ellos son los que usan esos paquetes diariamente es común que haya lanzamientos seguido con actualizaciones, parches y mejoras. Esto obviamente depende del desarrollador, pero por mi experiencia personal suele ser muy seguido.

2.No es relevante por el punto 1.

3.Dado que hay suficientes desarrolladores para empaquetar todo el software no hay prácticamente aplicaciones fuera de la distribución. La mayoría del software ya esta empaquetado para Debian. Si encontramos software que no lo este podemos recomendarlo o mantenerlo nosotros.

Preguntas frecuentes.

¿Qué es HURD?

Hurd es una colección de servicios que se ejecutan sobre un micro núcleo para implementar las distintas funcionalidades. El Hurd es el kernel oficial del proyecto GNU. Debian es la única distribución que ofrece el kernel de Hurd en su sistema operativo. Actualmente está disponible para testearlo en el sitio oficial de Debian y es un proyecto en activo desarrollo. Inicialmente se desarrolló porque no había otro kernel disponible, el kernel de Linux todavía no existía y había un montón de herramientas GNU que no tenían sistema operativo. Luego cuando Linux se hizo popular comenzaron a usarse estas herramientas con el kernel de Linux y nació el sistema operativo GNU/Linux. Tiempo después, Debian GNU/Linux.

¿Qué son los nombres woody, sarge, sid?

Son los codename de las versiones que se mantienen de la distribución. Están tomados de la película Toy Story. Por ahora se han utilizado los siguientes nombres: buzz (Buzz Lightyear), rex (el tiranosaurio), bo (Bo Peep, la nenita que cuidaba a la oveja), hamm (el chanchito alcancía), slink (Slinky Dog, el perro de juguete), potato era, por supuesto, Mr. Potato, woody (el cowboy) y sarge (el sargento del ejercito de soldaditos de plástico).

Cada vez que una versión en testing es aprobada, pasa a ser la versión stable y el nombre que usaba pasa a stable.

En su momento, woody fue testing y ahora es stable. Lo mismo va a pasar con sarge. La diferencia con sid es que siempre va a ser unstable. Sid era el nenito de la casa de al lado que rompía los juguetes :)

También existe un release más, experimental. En este release es donde las aplicaciones están disponibles en un principio y son paquetes que sencillamente fueron empaquetados en deb y jamás fueron probados. Es para gente que quiere estar muy a lo último.

Conclusión

En conclusión, el proyecto Debian nos ofrece un sistema operativo libre, robusto, plenamente probado, con un sistema de paquetes impecable y un equipo de desarrolladores dedicado a cumplir con los estándares GNU. Mientras que por otro lado nos deja abierta la posibilidad de otros sabores con versiones actuales en sus distribuciones testing y unstable. Como si eso fuera poco, tenemos disponible Debian para netbsd, freebsd y hurd.

Los Mejores Sitios Linux

A pesar del auge y la divulgación de Linux, hay personas y empresas que recién están empezando a preguntarse qué es, qué pueden hacer con él, cómo lo consiguen, etc... Los sitios que les mostramos hoy nos sacan de estas dudas, nos mantienen al día de las novedades y noticias sobre este tema, y algunos nos introducen en temas un poco más avanzados



Nombre: El rincón de Linux
URL: www.linux-es.com
Síntesis: Este sitio es un muy aconsejable para quien recién está empezando a incursionar en el tema. Tiene una breve explicación de qué es Linux, un poco de su historia, principales características, diferentes distribuciones, preguntas frecuentes, etc. Bien armado, liviano y simple de navegar, totalmente recomendable



Nombre: Linux para todos
URL: www.linuxparatodos.com
Síntesis: A pesar de su nombre, este sitio no se queda atrás a la hora de atrapar la atención de experimentados en la materia. Tiene una envidiable sección de manuales para descargar y una lista de foros de soporte muy actualizado.

HISPA LINUX

Nombre: HispaLinux
URL: www.hispalinux.es
Síntesis: Es el sitio de la Asociación Hispana de Usuarios de Linux. Interesante foro de referencia e interés para el mundo Unix en general y del software libre y GNU en particular.



Nombre:PenguinPPC
URL:www.penguinppc.org



Nombre:*I-MAC Linux*
URL:www.imaclinux.net

Síntesis: Ambos sitios están pensados para usuarios de plataforma Mac que quieren incursionar en sistemas operativos libres. En ambos se pueden encontrar varios niveles de dificultad en su documentación, así como noticias, desarrollo, y por supuesto los

Los Mejores Sitios Linux (cont.)

Otros Sitios Recomendables de Habla Hispana

- **Asociación Informática Abierta** http://www.abierta.org
- **AsturLiNUX** http://www.asturlinux.org/
- **Grupo de Usuarios de Linux en Maracaibo** http://mcbo.linux.org.ve
- **BULMA** http://bulmalug.net/
- **AQPLUG** **Grupo de Usuarios de Linux en Arequipa** http://linux.aqp.com.pe
- **Grupo de Usuarios de Linux Argentina** www.linux.org.ar
- **Grupo de Usuarios de Linux de Mendoza** www.linuxmendoza.org.ar
- **Grupo Linux de Occidente** www.glo.org.mx
- **Grupo de Usuarios de Linux de Perú** www.linuxperu.org
- **Grupo de Usuarios de Linux de Rosario** www.lugro.org.ar
- **Linux México** http://www.linux.org.mx/
- **Grupo de Usuarios de Linux de la Universidad Carlos III de Madrid** http://gul.uc3m.es/gul/princ.html
- **Grupo de Usuarios de Linux de El Salvador** www.linux.org.sv
- **GULEV** **Grupo de Usuarios de Linux en el Estado de Veracruz** http://veracruz.linux.org.mx
- **Grupo de Usuarios de Linux en Monterrey** www.interclan.net/rsepuiveda
- **Colectivo de Apoyo a Linux de Asturias** http://www.coala.uniovi.es
- **Gulex** http://www.gulex.org/
- **Grupo de Usuarios de Linux D.F** www.linux.org.mx
- **Grupo de Usuarios de Linux del ITESM CES** http://linux.cem.itesm.mx
- **Universidad Valle Linux en la Universidad Valle** http://cluv.univalle.edu.co
- **Linux en Puebla (México)** www.puebla.linux.org.mx
- **VELUG** **Grupo de usuarios Linux de Venezuela** www.linux.org.ve
- **UYLUG** **Grupo de Usuarios Linux en Uruguay** www.linux.org.uy
- **Grupo de Usuarios de Linux de Costa Rica** www.linux.or.cr
- **Spanish Linux Users Group** http:/ /slug.hispalinux.es
- **BALUG** http://www.balug.org.ar/
- **Grupo de LinuxUarios de Bizkaia** http://glub.ehu.es/index.html
- **G.P.U.L** Http://ceu.fi.udc.es/GPUL/

La Mejor Documentación

LCHispaLinux: http://www.hispalinux.es
INSFLUG: www.insflug.org
Linux Landia: http://www.linuxlots.com/-barreiro/tld.html
LuCAS: http://lucas.hispalinux.es
PAMELi: Http://ditec.um.es/-piernas/rnanpages-es
WGS Inc.
Linux Free Magazine: http://www.linuxmall.com
Barrapunto: Http://barrapunto.com
SuSE Hispano: Http://www.susehispano.org

Asociaciones en Red y Estándares

- **American National Standars Institute** http://www.ansi.org
- **ATM Forum** http://www.atmforum.com
- **Debian en español** http://www.es.debian.org
- **Proyecto GNU** http://www.gnu.org
- **Interest, Inc.** http://www.spi-inc.org
- **Linux Standard Base** http://www.linuxbase.org
- **Telecommunications Standards Institute** http://www.etsi.org
- **The Open Group** http://www.opengroup.org
- **The USENIX Association** Http://www.usenix.org

Recursos de Hardware y Red

Servicios hosting y servidores
http://www.bigfoot.com
http://www.webmastersnet.com
http://www.w3.org/Daemon

Sendmail
http://www.sendmail.org

Qmail
Http://www.qrnail.org

Empresas y distribuciones

Caldera www.caldera.com
Corel http://linux.corel.com
Enhanced Software Http://www.estinc.com
Intel www.intel.com
Linuxcare http://www.linuxcare.com
Linux for PowerPc http://www.linuxppc.org
MandrakeSoft http://www.mandrakesoft.com
netSweng, LLC http://www.netsweng.com
MSC Software http://www.msclinux.com
Oracle http://technet.oracle.com/tech/linux/content.html
Turbolinux Inc. http://www.turbolinux.com
Red Hat Software http://www.redhat.com
SuSE http://www.suse.de
Yggdrasil http://www.yggdrasil.com
Linux empresarial http://www.m-tech.ab.ca/linux-bizJ
Slackware Http://www.slackware.org

Seguridad

SecurityFocus Http://www.securityfocus.com
WindowMaker Http://www.windowmaker.org

¿QUIÉN PUEDE PROGRAMAR UNA APLICACIÓN,

corregir errores, atender a un cliente, migrar una base de datos y documentar un sistema al mismo tiempo? Un desarrollador, por supuesto. ¿Y quién puede ofrecerle una publicación para mantenerse actualizado, capacitarse, obtener recursos y conocer nuevas herramientas? **USERS .CODE**, por supuesto.

Finalmente llegó la publicación que la comunidad de desarrolladores estaba esperando, la revista que va a ocuparse de sus necesidades. Todos los lenguajes, todas las plataformas, proyectos, ejemplos, códigos, noticias, reviews, toolbox, white papers y las opiniones de los principales expertos.

Con **USERS .CODE** los desarrolladores compartimos el mismo código.



SUSCRÍBANSE Y RECIBIRÁN CON CADA EDICIÓN DE USERS .CODE UN COMPLETO CD-ROM CON MATERIAL SELECCIONADO Y TESTEADO POR NUESTROS EXPERTOS:

Aplicaciones | Demos | Compiladores | Librerías | Ejemplos | Código fuente | Cursos, videos y presentaciones | Y todas las herramientas que necesitan...

15% OFF P/SUSCRIPTORES DE USERS

AR
- Web: usershop.tectimes.com
- Teléfono: (011) 4959-5000
- Mail: usershop@tectimes.com



- Web: usershop.tectimes.com
- Teléfono: 55-5600-4815
- Mail: usershopmx@tectimes.com



Revisiones de exámenes LPI 101: Certificación Neutral Linux

Este artículo toma del examen LPI 101, los requisitos para rendir la certificación LPIC 1. (Ver Fig. 1)

El Linux Professional Institute es un instituto canadiense sin fines de lucro, creado en 1999 con el fin de desarrollar una certificación Linux de vendors neutrales patrocinada por una comunidad.

LPI actualmente ofrece dos niveles de certificaciones de Linux: Nivel 1 (LPIC 1) y Nivel 2 (LPIC 2). El otro, el nivel (LPIC 3), está actualmente en pleno desarrollo. La certificación LPIC 1 es un requerimiento recomendado para varias certificaciones Linux de vendors específicos, incluyendo las de IBM y Novell. Esta revisión de examen trata los temas del primero de los dos exámenes necesarios para obtener la certificación LPIC 1, el examen 101.

El examen 101 de LPI consiste en preguntas tipo multiple choice (múltiples opciones) y fill-in-the-blanks (completar los espacios). Note que el 101 no significa que el examen sea fácil: se espera que los candidatos de LPI tengan un sólido entendimiento de un amplio rango de tópicos, y LPI es famoso por hacer preguntas sobre opciones oscuras de comandos.

Objetivos de examen

El examen 101 de LPI está dividido en cinco principales áreas de objetivos:

- Hardware y Arquitectura
- Instalación de Linux y Manejo de Paquetes
- Comandos GNU & Unix
- Dispositivos, Sistemas de Archivos de Linux, Estándares de Jerarquía de Sistema de Archivos (Filesystem Hierarchy Standard-FHS)
- Sistemas X Windows

Los objetivos de los exámenes de LPI se presentan un poco diferentes a lo que usted seguramente está acostumbrado a ver.

Cada uno de los tópicos enumerados anteriormente, contienen sub-tópicos que están valuados en una escala de 1 a 10, siendo 10 el mayor valor. Preguntas relacionadas a tópicos más importantes aparecerán en mayor número que aquellas relacionadas o tópicos de menor importancia los objetivos de exámenes no están creados todos iguales.

Su punto de partida deberá ser una visita a al sitio web de LPI y leer todos los objetivos, prestando mucha atención a aquellos con nivel de escala 5 o más. (Mientras esté aquí, también preste mucha atención a las preguntas de ejemplo que provee LPI).

Para darle una mejor idea de lo que estamos hablando, a continuación haremos una descomposición de los 5 objetivos principales.

Hardware y Arquitectura

El enfoque hacia hardware de este examen tiende a ser bastante básico, como por ejemplo identificar tipos de placas o conectores, instalación de discos rígidos (SCSI), configuración de placas de sonido, como también instalación de módems. Familiarícese con:

```
/proc/ioports
/proc/interrupts
/proc/dma
/proc/pci
```

Conozca lo que estos archivos dinámicos contienen, qué información puede ser acumulada por ellos, y cómo se adaptan a instalaciones de hardware.

Por ejemplo: si usted piensa que un módem y una placa de sonido no estaban funcionando por causa de un problema con IRQ, puede ir al archivo **/proc interrupts** y posiblemente ver o incluso resolver este problema.

Tip: en la mayoría de los casos a Linux no le gusta los winmodems!

Instalación de Linux y Manejo de Paquetes.

LPI espera que sus candidatos del examen 101 sean capaces de instalar Linux en un sistema limpio. También requiere que sean capaces de instalar y/o actualizar software en ese sistema Linux después de la instalación inicial. Es costumbre de los verdaderos vendors neutrales de LPI no tener favoritos. La habilidad de instalar y actualizar software desde un código fuente, de rpm´s (Sistema de paquetes del Red Hat), y deb´s (Sistema de paquetes Debian) son absolutamente críticos para toda esperanza de pasar este examen. Conozca toda la línea de opciones de comandos para ambos manejos de paquetes, por ejemplo:

```
Rpm qip <package>
```

Este comando le proveerá información sobre un paquete rpm que usted haya descargado pero todavía no instalado en su sistema. No descuide a Debian si usted no lo usa, planee memorizar los comandos y sus opciones (generalmente Debian es clasificado como una distribución más compleja de Linux para usuarios avanzados), por ejemplo:

```
Dpkg r <package> or dpkg -remove
<package>
```

Sepa que esto removerá un paquete Debian, pero NO los archivos de configuración. Para remover completamente el paquete del sistema deberá correr:

```
Dpkg P <package> or dpkg -purge
<package>
```

Esto purgará los archivos de configuración del sistema, en este punto el paquete será completamente removido del sistema.

Saber usar el Administrador de paquetes Red Hat y Debian es un gran comienzo, pero también se espera que usted pueda compilar software desde su fuente usando las herramientas detalladas a continuación, las que se encuentran en muchas distribuciones:

```
tar
bzip2
gzip, gunzip
./configure
make
make install
```

Se espera que los candidatos sepan qué son las librerías compartidas y cómo trabajar con ellas (Ej.: **ldd, ldconfig, ld.so.conf, LD_LIBRARY_PATH)**.

Finalmente, los administradores de inicio de sistema (boot loaders) para Linux están enfatizados en estos objetivos, por eso, los candidatos deben estar familiarizados con LILO (LInux Loader) y GRUB (GRand Unified Bootloader), como también sus respectivos archivos de configuración lilo.conf y grub.conf.

GNU y Comandos Unix

Esta sección hace a la estructura del examen, y usted deberá dedicar gran parte del tiempo en la preparación de este objetivo. Al LPI le encantan las opciones de líneas de comando y evalúa firmemente sobre ellas. Un consejo: para este examen usted tiene que memorizar todas las opciones de línea de comando para los comandos en éstos objetivos.

Algunos de los tópicos y comandos con los que un candidato debe estar familiarizado son:

- Trabajar con procesos: **ps, nice, renice, top**
- Trabajar desde la línea de comando: **bash, set, unset, echo, export**, etc.
- Trabajar con el editor vi: **:q, :q!, :w, :w!, ZZ, dd, /, ?,** etc.
- Procesar y buscar textos: **sed** y **grep**
- Usar pipes y redireccionamiento: **|, tee, >, <, >>,** etc.
- Manejo básico de archivos**: cp, mv, ls, cat, rm, mkdir**, etc.

Dispositivos, Sistemas de Archivos Linux, Estándares de Jerarquía del Sistema de Archivos (Filesystem Hierarchy Standard-FHS)

Los candidatos al examen 101 deberán conocer el proceso de crear particiones en discos, crear sistemas de archivos en particiones, montar y desmontar sistemas de archivos, mantener y reparar esos sistemas de archivos, usar cuotas de espacio en disco, como también saber qué sistemas de archivos deben ser organizados.
Los sistemas de archivos en Linux siguen a un estándar llamado FHS (Filesystem, Hierachical Standard), el que básicamente sugiere cómo ciertos directorios deben estar organizados. Por ejemplo:

- Los archivos Log (especialmente logs de servidores web) y los que tengan el potencial de crecer rápidamente y desbordar un disco rígido, deberían ser colocados en algún lugar en el directorio /var o subdirectorio del mismo, y ese directorio /var usualmente está en su propia partición.
- Los directorios de los usuarios generalmente están en el directorio /home/<username>, y ese mismo directorio se debe colocar en una partición separada para evitar que los usuarios llenen su disco rígido.
- Este objetivo también está relacionado a los permisos de archivo o nivel de grupo y usuario, para la búsqueda de archivos en el disco rígido y la creación de enlaces (simbólico y duro). Ud. debería probar de cambiar la propiedad o los permisos de un archivo, buscar un archivo, tanto en su ruta de acceso permitido como fuera de ella, y el porqué debería o necesitaría crear enlaces de éstos tipos.

Tip: Recuerde que los enlaces duros no pueden migrarse entre sistemas de archivos, porque el enlace comparte el inodo (puntero) con el archivo original. Si necesita dividir sistemas de archivos, deberá utilizar un enlace simbólico.

El Sistema X Windows

El área de objetivos final en el examen 101 es el Sistema X Windows para Linux. La clave para esta sección es aprender y memorizar los archivos de configuración involucrados con la instalación, configurar y correr X Windows, como también tener clientes remotos conectados a una sección X window en un "server".

Estos archivos incluyen:
.xinitrc (note el "**.**" delante de cada archivo, esto indica que es un archivo oculto)
.Xdefaults
.Xresources
XF86Config (note que éste es el archivo de configuración, usted usa el comando "**xf86config**" -en minúsculas- para configurarlo).

Algunos comandos que definitivamente tiene que conocer son**: XF86Setup, xvidtune, xf86config** y **xhost.**

Tip: Sepa cómo configurar un entorno de escritorio y cómo configurar su display manager, como por ejemplo XDM (X Display Manager), GDM (Gnomo Display Manager) ó KDM (KDE Display Manager).

Estudiando para el examen

Muchas personas que han tomado ya este examen, el año pasado por ejemplo, no han podido encontrar ni un sólo libro que cubra los objetivos antes descriptos.

Y aquí la mala noticia: Esta situación no ha cambiado mucho desde ese entonces. Muchos terminaron haciendo lo siguiente: usar dos libros viejos:"LPI Linux Certification in a Nutshell" de O´Reilly y "The LPI Certification Bible" de John Wiley e Hijos. Luego tomar los objetivos del examen y marcar todos los capítulos que necesitaba leer.

Esto funcionó, pero andar saltando de un lado a otro en un libro da un poco de vértigo. La información en éstos dos libros lo llevará a través de este examen, pero *solamente* si usted ha instalado y usado Linux.

Es imposible enfatizar lo suficiente sobre lo importante que es para usted instalar Linux (varias veces) y hacer las actividades que le enseñan estos libros.

Si usted es nuevo en Linux, le recomiendo que use SuSE Linux o Mandrake Linux son relativamente fáciles de usar y son muy buenos para ayudarlo a prepararse para un examen.

Ideas finales

El examen 101 de LPI es un verdadero desafío, y aunque pregunte demasiado sobre opciones de línea de comando, es un examen muy aceptable. Las preguntas "fill-in-the-blank" requieren mucha concentración -asegúrese de tipear todo lo que se le pide (LPI indica cuando quiere cosas como, por ejemplo, opciones de comando).

Cualquiera que quiera demostrar que tiene habilidades moderadas o de administrador de sistemas "junior" en Linux, tiene que obtener la certificación LPIC1.

Figura 1

Examen	#117-101: examen LPI 101 (uno de los dos exámenes requeridos por la Certificación LPIC Nivel 1
Vendor	Linux Professional Institute (LPI)
Estatus	Live (actualizado en Marzo del 2003).Disponible en los centros de examen Prometic y Pearson Vue on line y en eventos patrocinados por LPI.
Rating	"No deje que el 101 lo engañe: este examen NO es para principiantes en Linux, requiere por lo menos una moderada experiencia con el OS Linux"
Información sobre el examen	Aproximadamente 65 preguntas, 90 minutos como tiempo límite, se necesita un puntaje de 500, en una escala de 200 a 890, para aprobar. Costo U$s 100.-
¿Quién debe tomar este examen?	Candidatos que tengan práctica en Linux y quieran probar sus capacidades en administración de sistemas. NOTA: Deben tomar también el examen 102 para obtener la certificación de Nivel 1 de LPI.